A QUATERNION APPROACH TO THE MODAL ANALYSIS AND REDUCED-ORDER MODELING OF THREE-DIMENSIONAL FLUID SYSTEMS

A Dissertation
Presented to
The Academic Faculty

By

Yanal Issac

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology

August 2020

Copyright © Yanal Issac 2020

A QUATERNION APPROACH TO THE MODAL ANALYSIS AND REDUCED-ORDER MODELING OF THREE-DIMENSIONAL FLUID SYSTEMS

Approved by:

Professor Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Professor Lakshmi Sankar
School of Aerospace Engineering
Georgia Institute of Technology

Doctor Olivia Pinon-Fischer
School of Aerospace Engineering
Georgia Institute of Technology

Doctor Walter Silva
Aeroelasticity Branch
NASA Langley Research Center

Doctor Jenna Eppink
Flow Physics and Control Branch
NASA Langley Research Center

Date Approved: July 1, 2020

the heart knows reasons of which reason knows nothing of

Blaise Pascal

Dedicated to my mother and father.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Dimitri Mavris, for his unwavering support and patience that cannot be underestimated. I would also like to extend my gratitude to my mentor at NASA Langley, Dr. Walter Silva, for his insightful suggestions and profound belief in my abilities. Over the past few months, Dr. Olivia Fischer relentlessly helped me with editing my thesis. Her invaluable contribution was instrumental in the writing of this thesis to which I am deeply indebted to. I also want to extend my gratitude to Dr. Jenna Eppink, for allowing me to take part in her experiments and for her constructive criticism and advice. Thanks should also go to Professor Laksmhi Sankar for his helpful advice and practical suggestions.

During my time as a Ph.D. student, my advisor Professor Dimitri Mavris arranged for me the opportunity to complete my studies at the NASA Langley Research Center. For that opportunity, I am truly appreciative of my advisor. My time as NASA proved to be an invaluable experience as it gave me the great opportunity to work alongside the researchers and scientists at the Transonic Dynamics Tunnel (TDT). It was there where I met my mentor Dr. Walter Silva, and had the chance to rub shoulders with him and many other bright researchers. It was a great pleasure of mine to work alongside Dr. Kevin Jacobson and Thomas Ivanco who had a positive influence on me. But I am particularly grateful to Pawel Chwalowski, for his assistance with FUN3D and for his great conversations at the break room where we spoke about life, science, and martial arts. That being said, I would also like to thank my friend Alexandre Damiao for his company and the good times we spent spent together.

I dedicate this thesis to my mother, who by example taught me to the meaning of the words loyalty and courage, and to my father, who taught me honor and to never give up.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Acronyms

NASA National Aeronautics and Space Administration

TDT Transonic Dynamics Tunnel

CFD Computational Fluid Dynamics

ROM Reduced-Order Model

MOR Model-Order Reduction

PDE Partial Differential Equation

ODE Ordinary Differential Equation

POD Proper Orthogonal Decomposition

QPOD Quaternion Proper Orthogonal Decomposition

ERA Eigensystem Realization Algorithm

ML Machine Learning

Number Sets

$\mathbb{R}$ Real Numbers

$\mathbb{C}$ Complex Numbers

$\mathbb{H}$ Quaternion Numbers

Symbols

u	Velocity in the x direction
v	Velocity in the y direction
	Velocity fluctuation vector about the mean
w	Velocity in the z direction
U	Velocity vector
U^{q}	Quaternion representation of the velocity vector
ρ	Density
ν	Kinematic viscocity
e	Internal energy
n	Dimension of full-order model
m	Dimension of reduced-order model
N	Number of snapshots
x	State vector of the full-order model
$\dot{x}$	State velocity vector of the full-order model
a	State vector of the reduced-order model
$\dot{a}$	State velocity vector of the reduced-order model
R	Two point spatial correlation function
R^{d}	Discrete two point spatial correlation matrix
U^{d}	x component of the data

V^{d}	y component of the data
W^{d}	z component of the data
u^{d}	Snapshot matrix
u_m^{d}	rank-m approximation of the snapshot matrix
u^{dq}	Quaternion snapshot matrix
u_m^{dq}	rank-m approximation of the quaternion snapshot matrix
ϕ	Coherent structure (POD mode)
ϕ^u	x component of coherent structure
ϕ^v	y component of coherent structure
ϕ^w	z component of coherent structure
ϕ^{q}	Quaternion coherent structure (QPOD mode)
ϕ^{rq}	real component of quaternion coherent structure
ϕ^{iq}	i-th imaginary component of quaternion coherent structure
ϕ^{jq}	j-th imaginary component of quaternion coherent structure
ϕ^{kq}	k-th imaginary component of quaternion coherent structure
S	Subspace spanned by basis functions
S^{q}	Subspace spanned by quaternion basis functions
λ	Eigenvalue
λ^{q}	Quaternion eigenvalue
σ	Singular value

σ^{q} Quaternion singular value

$\theta(\lambda^{\mathrm{q}})$ Similarity orbit

$\kappa(\lambda^{\mathrm{q}})$ Kaleidoscope set

SUMMARY

In 1967, Lumley [1] derived the proper orthogonal decomposition (POD) in an attempt to provide a mathematical description of patterns that emerge in turbulent flow, which he called coherent structures. The POD method derived by Lumley has deep mathematical roots, is statistically based, and has analytical foundations that provide a rigorous mathematical framework for the extraction and description of coherent structures [2].

However, it was not until 1987, when Sirovich [3] provided a numerically tractable implementation of the POD method capable of tackling large data sets such as the ones encountered in modern day engineering problems, did the POD gain much attention. In recent years, the wide spread success of the POD method has greatly elevated the status of the POD method. Today, the POD method is considered as benchmark procedure, and is at the bedrock of modal analysis and reduced-order modeling of fluid systems [4].

Since its inception, numerous applications, adaptions and variations of the POD have been devised. However, little attention has been paid to addressing the three-dimensional nature of fluid systems. In fact, George [5] states that the POD is agnostic to the nature of the data, as it does not matter whether the data is velocity, pressure or temperature.

The aim of this thesis was to explore a fundamentally different approach to the POD that is better suited for three-dimensional fluid systems; an approach that does not compromise the mathematical rigor associated with the concept of coherent structures defined by Lumley. The approach investigated in this thesis replaces the traditional field of real numbers $\mathbb{R}$, with a four-dimensional non-commutative division algebra $\mathbb{H}$, known as the quaternion division algebra.

To the knowledge of the author, this thesis is the first to incorporate quaternions into Lumley's mathematical framework. The introduction of quaternions into Lumley's mathematical framework, generalizes the proper orthogonal decomposition to the quaternion proper orthogonal decomposition (QPOD) while preserving its favorable features and ex-

tending the POD to higher dimensional spaces.

In the work of this thesis it was shown that a quaternion approach abstracts Lumley's mathematical representation of coherent structures at a fundamental level. These abstracted representations, defined for the first time in the work of this thesis are termed quaternion coherent structures, exhibit interesting properties and result in a fascinating phenomena termed the kaleidoscope effect which is not present in the traditional definition of coherent structures. Furthermore, it was numerically and mathematical shown that the QPOD method can better distill the essential dynamics present in a data set and can create more accurate rank-m approximations as compared the POD method.

The results presented in this thesis provide compelling evidence advocating for the use of quaternions in the context of modal analysis and reduced-order modeling of three-dimensional fluid systems. In addition a numerical implementation of the QPOD inspired by the work of Sirovich [3] is also presented. The numerical implementation is termed the quaternion snapshot method and in the work of this thesis is shown to be scalable to large systems.

Hence, a quaternion representation of the velocity flow field variables, provides a more natural means of incorporating the flow variables into a single holistic variable, which addresses the three-dimensional nature of the data. Such a quaternion representation provides for a more natural, physics-based framework for the treatment of three-dimensional fluid systems which results in more informative modal analysis and more accurate reduced-order models.

The consequences of an improved modal analysis of fluid systems will greatly help scientists and engineers further their understanding of fluid flow. In addition, the QPOD method provides a superior capability to capture, isolate, and distill the complex aerodynamics resulting in faster, and more accurate reduced order models which will aid in many aspects of aircraft analysis and design, particularly aeroelastic analysis and design. These accurate but lower-order representations will also pave the way for surrogate-based

optimization, uncertainty quantification, and fluid flow control over flexible structures encountered in modern day and future aircraft designs.

CHAPTER 1
INTRODUCTION

1.1 Motivation

Two recent trends have emerged that are greatly influencing the future requirements of commercial aircraft design, namely, the need for faster and more flexible aircraft [6, 7, 8, 9]. Moreover, aircraft designers are typically motivated to use less structural material, as it leads to lighter, faster, more fuel efficient, but also more flexible aircraft. In order to meet the demands of future aircraft designs, highly flexible wings will be crucial [10]. This increase in aircraft flexibility renders the aircraft more vulnerable to problems such as flutter, buffeting, gust response, and many others, all which need to be accounted for during design [11].

Aircraft design, by its very nature, is an interdisciplinary, iterative and convoluted process. To surmount the complexities of aircraft design, aircraft designers and engineers have developed methods that heavily rely on trending historical data from previously built aircraft. For more details, Raymer [12] and Anderson [13] delve into great depths of the aircraft design process. These traditional aircraft design methods assume a rigid aircraft structure in the conceptual design and the preliminary design stages. At the early stages of design, aeroelasticity is accounted for using empirical estimates from similar, previously built aircraft. It is not until the latter stages of the design process, particularly the detailed design stage, that the assumption of rigidity is relaxed and flexibility is introduced, allowing for the aeroelastic analysis of the aircraft [14]. Modern aeroelastic analysis has evolved by examining how the unsteady aerodynamics and the flexible aircraft structure interact by formulating both an aerodynamic model and a structural model, and then coupling both models together [15, 16, 17, 18, 19, 20]. In order to build both the aerodynamic model and

the structural model, a great degree of knowledge of the flight conditions (Mach number, angle of attack, etc.) and of the aircraft's design itself (geometry, material properties, etc.) need to be known. It is for these reasons that aeroelastic analysis is performed at a later stage of the design process [14].

However, as new configurations, concepts and materials are being explored, the traditional aircraft design methods fail due to a lack of relevant historical data to trend [21]. Moreover, assuming rigidity for an inherently flexible design at the early stages of design dismisses the added potential during design space exploration associated with flexible structures, which consequently results in sub-optimal designs [14]. As aircraft designs migrate towards more flexible structures that operate in the subsonic, transonic, and supersonic regimes of flight, aeroelastic analysis becomes more detrimental and influential on the aircraft design [6]. Indeed, increasing the aircraft structural flexibility increases the difficulty in analyzing and designing the aircraft due to the highly nonlinear coupling between the structure and the unsteady aerodynamics, specially in the transonic regime [6]. In short, the traditional aircraft design process is inadequate for designing the inherently more flexible aircraft needed to meet the challenges of the future.

Designing the flexible aircraft of the future is going to require the capability to perform accurate, physics-based, trade-off studies that take aeroelasticity into consideration at the earlier stages of the design process [6]. Recent advancements in computing power and computational fluid dynamics (CFD) have made it possible to digitally duplicate and model aeroelastic effects at different flight and design conditions. These advancements mean that engineers are now able to investigate and analyze their designs via computer modeling and simulation, resulting in huge cost and time savings associated with prototyping, wind tunnel testing and flight testing.

Unfortunately, in many instances, the amount of computational resources (CPU time and memory) required to simulate a single configuration, at a single operating condition, can be demanding, hence limiting the design role of CFD to a few point solutions [11, 6].

This renders CFD impractical for applications where a large number of function evaluations is required, such as design space exploration, parametric studies, uncertainty quantification, optimization, etc. Yondo, Andrés, and Valero [22] explain how the deficiency in computational resources, particularly CPU time, raises difficulties for aircraft engineers in two different settings. Firstly, a many-query setting, which would enable aircraft engineers to perform studies that require large amounts of complex, time consuming, function evaluations for different aircraft configurations and design variables, paving the path for optimization, wider design space exploration, uncertainty analysis, sensitivity analysis, etc. Secondly, a real-time setting, which would enable aircraft engineers to predict an aircraft's behavior instantaneously, paving the path for the creation of interactive flight simulators, flow controllers, real-time estimators of aerodynamic coefficients, etc. Furthermore, Lucia, Beran, and Silva [23] explain how in spite of the physics-based CFD models producing large amounts of detailed data, very little insight and understanding of the physics governing the underlying phenomenon is gained by the aircraft engineer.

To address these issues, the scientific community has directed more attention towards reduced-order modeling, where a high-fidelity tool is approximated by a lower-order model that runs much faster. Using the techniques of model-order reduction [24, 25, 26, 27, 28], the essential dynamics of the higher-order model is captured and distilled into a cheap-to-evaluate, lower-order model. This lower-order model serves as a surrogate to the full blown, high-order CFD model. These cheap-to-evaluate reduced-order models (ROMs) provide aircraft engineers the means to run a lot more model simulations in a much shorter period of time, hence enabling them to perform the necessary multidisciplinary studies needed during the aircraft design process. Many times, the reduced-order models run orders of magnitude faster than their higher-fidelity counterpart, which allows for the real-time evaluation of these models and makes them suitable for real-time applications such as simulators and flow controllers. Furthermore, since reduced-order models isolate and simplify the dynamics of the phenomenon being examined, they also provide intuition into

the underlying principles governing the phenomenon at hand. In the scientific community, reduced-order models are also called surrogate models. Most of the attention of the reduced-order modeling community is dedicated to the model reduction of the unsteady aerodynamics. This is because the numerical computation of the unsteady aerodynamics is more time consuming and more difficult to predict. The application of model reduction to unsteady aerodynamics and fluid flow is well documented in the literature [29, 30, 4, 23]. A survey of surrogate modeling for aircraft aerodynamic analysis and design optimization can be found in references [31, 32, 22]. According to Bhatia [14], three model-order reduction methods have shown promising results in aeroelastic analysis. These include the p-transform method [33], the proper orthogonal decomposition [34] (also known as the Karhunen-Loève) method, and the Volterra methods [35].

Different model reduction methods are more suitable than others depending on the situation at hand. For example, some model reduction methods are more useful when the underlying governing equations of the problem are unavailable or inaccessible. Such methods are referred to as black box methods as they solely rely on observational data. Black box methods in the scientific literature are also known as data driven methods, and have received considerable attention in other scientific communities. White box methods tend to perform better than black box methods, however, unlike the black box methods, they are only applicable when the equations of the full-order system are available. Many of these techniques have been developed and studied in different research communities. For example, techniques for model-order reduction can be found in the fields of machine learning, data compression, dimensionality reduction, system identification and statistical learning [36].

Different circumstances and different applications dictate the model reduction method to be used. A wide range of model-order reduction methods typically follow a two-step procedure. In the first step, a set of basis functions is computed or chosen that satisfies a certain criteria. In the second step, a reduced-order model is generated using the resulting

basis functions from the first step. The proper orthogonal decomposition (POD) method in particular has received the most attention and is ubiquitous in all the previously mentioned fields, making it a benchmark method for generating reduced-order models.

In 1967, Lumley [1] mathematically showed that the POD basis functions, also known as POD modes, satisfy an optimality criterion for identifying emerging patterns in turbulent flows. Furthermore, many times a high dimensional systems evolves around a low dimensional attractor. The POD basis functions span the low dimensional attractor, hence identifying and approximating the attractor, which greatly reduces the complexity of the high dimensional system [2]. In 1987, Sirovich [3] published the snapshot method, a numerical implementation of the POD method, which has shown to be computationally efficient and scalable to very large systems. After the POD modes are computed, the next step is to generate a reduced order model. The most common method is the Galerkin projection [3] of the Navier-Stokes equations onto the subspace spanned by the POD basis functions computed in step one. Within the reduced-order modeling community, the POD-Galerkin projection method has become the center of attention, particularly for the fluid flow reduced-order modeling. When the underlying equations are large, complex, or unavailable, the POD method pairs well with many system identification and machine learning techniques such as the eigen-system realization algorithm (ERA) [37], neural networks [38], etc. One of the features that makes the POD method attractive is its versatility, i.e. the POD method can be used to create reduced-order models for both black box and white box scenarios. It is for all these reasons that the POD has gained its popularity and has become a staple in the reduced-order modeling community.

Unfortunately, there are several limitations associated with the POD method. A common problem with the POD method in fluid systems is the strong parameter dependence of the POD basis functions [39, 40]. In order to approximate the full-order model, the POD method uses numerical data from simulation runs at set parameters. The parameters typically represent a specific aircraft configuration or specific flight conditions (Mach

number, angle of attack, etc.) that can be varied. The optimal basis functions (POD modes) generated via the POD method tend to perform very well when the parameters are fixed, however, in many cases, when the parameters are perturbed, the POD basis functions fail to provide a good approximation of the flow. Another common limitation associated with the POD method is the stability of the reduced order models generated via the POD modes. The POD-Galerkin procedure guarantees that the resulting reduced-order model is stable if the full-order model is linear and stable, however, such guarantees do not hold for nonlinear systems. For example, in turbulent flow, when creating a ROM via the POD method, the high-energy modes are retained and the low-energy modes are truncated. However, the low-energy modes typically contain the small scale eddies responsible for energy dissipation. Therefore, ROMs constructed using a subset of the POD modes can be unstable even though the higher order model sampled is stable. Taira et al. [4] also note that the POD arranges the modes in order of energy content and not dynamical importance.

In other cases, the POD method has been shown to be inefficient in representing fully turbulent flows [41] as a large number of modes were required to produce a realistic ROM. It is not clear how many POD modes to keep and how many to truncate, as there are many truncation criteria [4]. Lumley first derived the POD method in order to extract what he referred to as coherent structures in turbulent flow. Turbulence, which seems to be a stochastic process, gives rise to identifiable, sporadically appearing structures, such as eddies, that greatly influence the evolution of the flow. Lumley named those intermittently appearing patterns as coherent structures, and was motivated in mathematically describe those structures which ultimately led to a derivation of the POD method. However, in many instances the POD modes are not easy to interpret and visualizing the POD modes does not add insight to the understanding of the underlying physics of the flow.

A great limitation of the POD method is due to the linearity of the method [2], i.e. the POD method can only capture the dynamics that can be expressed as linear combination of the data set collected. Furthermore, the POD does not take into account the underlying

dimensionality of the fluid system. Instead, the POD method treats one, two, and three dimensional problems in an identical fashion; by stacking the data of every time iteration into a single column. Tiara et al. [4] also note that the POD method relies on second-order correlations and ignores higher-order correlations. They also add that the POD method is not suitable in describing traveling wave structures (i.e. $f(\eta - ct)$), since those structures cannot be separated between temporal and spatial functions.

This work extends and builds upon the previous ideas suggested by Lumley [1] and Sirovich [3] and attempts to address some of the limitations associated with the POD method. To the best knowledge of the author, this thesis introduces an approach that generalizes the POD method to three-dimensional fluid systems in a fashion not done before. This novel approach provides a more informative modal decomposition of fluid systems which may help scientists and engineers further their understanding of fluid flow. Moreover, by improving the POD method, more efficient and more accurate reduced-order models of complex systems can be created. This enables the application of surrogate based methods (optimization, design space exploration, controls, etc.) needed to address the aeroelastic design problems of the future.

Research Objective. *Explore a new approach that addresses the three-dimensional nature of fluid systems in the context of modal analysis and reduced-order modeling.*

The second chapter of this thesis reviews the proper orthogonal decomposition, its derivation, and limitations. In the third chapter, the novel approach suggested by this thesis is presented, along with its necessary background information. The fourth chapter details the experiments carried out to assess and compare the POD method with the proposed method. The compelling results of the experiments, and a discussion of them is presented in Chapter five. Finally, chapter six concludes on the findings of this research, and note on the recommendations for future work is given.

CHAPTER 2
BACKGROUND

This purpose of this chapter is to provide an in depth discussion of the proper orthogonal decomposition. Section two motivates the need for model-order reduction in aircraft analysis and design. Section three provides a brief overview of model-order reduction, however, more attention is paid to projection based model-order reduction techniques. Section four introduces the proper orthogonal decomposition. Section five discusses POD based reduced-order models, their extensions, and limitations.

2.1 Historical Comment

A mathematical model of a system or a phenomenon is built to explain a system's behavior and to predict its future behavior. A mathematical model of a system can also be used to optimize the behavior of a system under certain performance criteria or to create controllers that would regulate the behavior of the system. Typically, in order to create a mathematical model, physics-based principles that explain the nature of a system or the phenomenon under consideration are applied, resulting in a set of equations that relates the different variables governing the system or phenomenon. As the complexity of the system or phenomenon increases, so do the equations that model it.

In the case of a moving fluid, three conservation principles govern its behavior:

1. Conservation of mass (continuity equation)
2. Newton's second law (momentum equation)
3. Conservation of energy (energy equation)

When expressed mathematically, these principles manifest themselves either as integral equations or as partial differential equations (PDEs). There are four useful constructs where

the previously mentioned principles can be applied, which result in a mathematical model for fluid flow:

1. Finite control volume fixed in space
2. Finite control volume moving with the fluid (fixed mass)
3. Infinitesimal fluid element fixed in space
4. Infinitesimal fluid element moving along a streamline

The application of the three governing principles to the four mentioned constructs gives rise to different but equivalent mathematical representations of the principles that govern fluid flow.

The velocity vector of a fluid at a given location and time is given by

$$U(x,y,z,t) = u(x,y,z,t)\boldsymbol{i} + v(x,y,z,t)\boldsymbol{j} + w(x,y,z,t)\boldsymbol{k} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \tag{2.1}$$

where the x, y, and z components of the velocity vector are given by u, v, and w respectively. Applying the conservation of mass principle to a finite control volume with volume V and surface area S fixed in space, gives the integral form of the continuity equation

$$\frac{\partial}{\partial t}\iiint_V \rho \,\mathrm{d}V + \iint_S \rho U \cdot \,\mathrm{d}S = 0, \tag{2.2}$$

where $\rho = \rho(x,y,z,t)$ denotes the density of the fluid at a specific location and time. Applying Newton's second law to a moving fluid element results in the Navier-Stokes

equations in the conservation form given by

$$\begin{aligned}
\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u U) &= -\frac{\partial p}{\partial x}, \\
\frac{\partial(\rho v)}{\partial t} + \nabla \cdot (\rho v U) &= -\frac{\partial p}{\partial y}, \\
\frac{\partial(\rho w)}{\partial t} + \nabla \cdot (\rho w U) &= -\frac{\partial p}{\partial z},
\end{aligned} \tag{2.3}$$

where $p = p(x, y, z, t)$ denotes the pressure at a specific location and time. Similarly, applying the conversation of energy principle results in the energy equation given by

$$\rho \frac{\mathrm{D}}{\mathrm{D}t}\left(e + \frac{\|U\|^2}{2}\right) = \rho\dot{q} - \frac{\partial(up)}{\partial x} - \frac{\partial(vp)}{\partial y} - \frac{\partial(wp)}{\partial z}, \tag{2.4}$$

where $e = e(x, y, z, t)$ represents the internal energy due to molecular motion at a specific location and time. These five equations, in terms of six unknown flow-field variables ρ, p, u, v, w, e, are a coupled system of nonlinear integral and/or partial differential equations. Unfortunately, this system of equations, which constitutes the governing equations of a single phase fluid flow, has no analytical, closed-form solution, even for some of the simplest of configurations. Consequently, due to the mathematical complexity of the physics-based models, analytically solving the Navier-Stokes equations is not possible.

Hence, some of the earliest physics-based models relied on simplifying assumptions in order to make the models more mathematically tractable. For example, early models of the unsteady aerodynamic response relied on the principles of superposition and the convolution of fundamental responses for two dimensional airfoils in incompressible flow. Under those assumptions, Wagner [42] introduced one of the earliest unsteady aerodynamic models known as the Wagner's function for modeling the unsteady aerodynamic response to a step change in angle of attack for a thin two dimensional airfoil. Using the principle of superposition, it is possible to compute the unsteady aerodynamic response to any change in angle of attack using convolution. Similarly, the Kussner's function [43] models the unsteady aerodynamic response to sharp-edged gust. Theodorsen's function [44] was later

derived for the analytic computation of the aerodynamic response due to sinusoidal motion for a two-dimensional airfoil in incompressible flow. Sear's function [45] extends the Kussner's function to model the response to a sinusoidal gust. These physics-based models are analytically-derived unsteady aerodynamic responses that are traditionally applied in the modeling of unsteady aerodynamics and aeroelasticity. Typically, these models assume:

1. Inviscid flow
2. Irrotational and isentropic flow
3. First order interaction between the flow and structure
4. Superposition

which allow the use of linear methods. These assumptions were adequate for early aircraft designs, particularly for aircraft designed to fly at low Mach numbers with a high aspect ratio.

However, as the need for faster, more maneuverable aircraft arose, the previously mentioned physics-based models fail. As some of these assumptions are relaxed, and as the geometric complexity is increased, the analytic derivation of the previously mentioned functions becomes more impractical. In actuality, aerodynamic flow is:

1. Three dimensional
2. Viscous
3. Compressible
4. Nonlinear

Insisting on an analytical-based solution severely limits the potential use of physics-based models in design.

Fortunately, it is still possible to use the underlying mathematical formulation of the physics-based models to find approximate solutions instead of exact ones. Due to advances

in computational fluid dynamics, it is possible to use numerical methods to find approximate solutions to unsteady aerodynamic responses for three dimensional geometries. One of the earliest advances in numerical methods that allowed for the unsteady aerodynamic modeling of three dimensional geometries is strip theory which originated in 1942. However, in 1969, the doublet lattice method [46] was developed by Albano and Rodden and proved to be superior over its predecessor. In the subsonic regime, the use of the doublet lattice method is still ubiquitous, while the use of piston theory in the supersonic regime and at small angles of attack is more suitable [47]. The previously mentioned numerical methods are linear, allowing for the use of the superposition principle, which makes these methods computationally fast and produce accurate results in their respective flight regimes. However, due to the highly nonlinear flow patterns in the transonic regime, the previously mentioned linear methods are inadequate in predicting the unsteady aerodynamics in the transonic regime, shifting the focus of the scientific community and design engineers to computational fluid dynamics. The increase in computational resources and the advancements in CFD have allowed engineers to perform detailed analysis on complex geometries that were not possible before. An important application of the CFD codes is in predicting the highly nonlinear effect of buffeting and flutter. An example of aeroelastic analysis done via CFD applied to a complete F-16 configuration at different Mach numbers in the transonic regime can be found in references [19, 48]. It took a 128-processor computing system to perform the analysis in less than one day for a single configuration.

Performing parametric studies is computationally taxing and intractable as the number of operating conditions and design variables increase. In order for engineers to properly design aircraft with complex geometries under various flight conditions, engineers need to perform a wide range of experiments at different operating conditions and under different design parameters and aircraft configurations. Examples of typical operating conditions of interest are free stream velocity, Mach number, angle of attack, altitude and Reynolds number, just to name a few. Such parametric studies are not currently possible due to the time

and memory limitations of the state of the art computational resources available. In short, engineers are faced with a design problem that requires time-consuming, highly coupled, nonlinear simulations for a large number of parameters, which limits the applicability of CFD codes and simulators.

Since CFD problems with moderate complexity can be computationally expensive, even on the fastest super computers, a reduced model that can run much faster and closely approximate the original high-fidelity model is sought after. This is known as reduced-order modeling. A reduced-order model of a phenomenon or a system is defined as the lowest dimensional model that can capture the dominant behavior of the given phenomenon or system. Recent advancements in machine learning algorithms, signal processing and big data analytics have made reduced-order modeling a possibility and a viable solution to surmount CFD shortcomings [49, 50, 51, 52]. Reduced-order modeling is expected to enable engineers, not only to perform parametric studies that would help analyze aerodynamic phenomena [51, 52, 23], but also perform optimization [53, 31, 32], controls [54, 55, 56], and when coupled with a structural model, aeroelastic and aeroservoelastic analysis at speed orders of magnitudes faster than CFD. For example Lieu, Farhat, and Lesoinne [39] applied the POD reduced-order model on a full F-16 fighter configuration to perform aeroelastic analysis at the transonic regime. The full-order aeroelastic model has over two million degrees of freedom and was effectively reduced to ninety degrees of freedom with less than 10% error. The authors also report a speed up of the linearized flutter analysis by a factor of five. It would also enable engineers to perform stochastic analysis, uncertainty analysis and multi-scale modeling. Furthermore, reduced-order models can also be applied when developing real time simulators to train pilots [33]. Applications of reduced-order models are many, and they appear in different forms in the various fields of science and engineering. In the next section a brief overview of model-order reduction is given. However, projection-based model-order reduction is emphasized as these methods have shown great success in describing flow mechanics and its techniques generalize to nonlinear sys-

tems [30].

2.2 Model-Order Reduction

A pervasive trend throughout the sciences, specially in engineering, is the tendency to describe and simplify complex entities into a combination of its more fundamental and elementary components. For example, complicated polynomials are factorized as a product of their roots, matrices are decomposed as the product of other simpler matrices, and complicated functions are expanded using tools like the Taylor series expansion, Laurent series expansion, power series expansion, Fourier series, etc. All these techniques attempt to break down obscure mathematical notions in terms of their simpler and more manageable counterparts. This aids scientists and engineers in developing a deeper understanding and an intuition into the behavior of what is being investigated. These techniques also provide another added benefit to the engineer, namely a method for approximation and model reduction. The literature on model reduction is large and extensive, several books have been written on the field, and only a brief overview is attempted here. The interested reader should refer to references [25, 26, 28] for an in depth treatment of the subject. A survey of model reduction methods can be found in reference [24].

Schilders et al. [26], provide an interesting perspective on model-order reduction in mathematics which is borrowed here. In 1807, Fourier published the idea of approximating a function with a few trigonometric terms. In linear algebra, Lanczos and Arnoldi made ground breaking progress in approximating matrices. In the area of dynamical systems and controls, systems are usually expressed in state space form. The order or complexity of such models is determined by the dimensionality of its state space. In this case model-order reduction is the attempt to find a less complex model that preserves the input-output relationship of the original model as much as possible.

Since state space models have infinite number of representations under similarity transformations, model-order reduction attempts to reduce the complexity of models by preserv-

ing the dominant characteristics of properties that are invariant under similarity transformations, such as the system's poles or the system's moments. Most techniques of model-order reduction attempt to retain one or more of the those invariant characteristics.

The next section, describes how aircraft engineers arrive at high-order models and the needs associated with simulating such models. This outlines the role of model-order reduction and its potential contribution in aircraft design.

2.2.1 High-Order Models in Aircraft Design

Model-order reduction is an active area of research that investigates the reduction of the complexity of a dynamical system while preserving the system's input and output behavior as much as possible. Model-order reduction investigates replacing ordinary differential equations that contain variables and/or equations in the order of 10^9 or more with a simpler, more manageable set of ODEs with much fewer variables and/or equations, while preserving the response characteristics and behavior of the original set of ODEs. By casting the ODEs into a state space form, complexity can be defined as the dimensionality of the state space. ODEs with a high-dimensional state space are ubiquitous in many areas of engineering, particularly in modeling and simulation applications. For example, modern day CFD models are notorious for being high-dimensional and exhibiting complex and nonlinear behavior. Simulating models with such complexity can be a daunting task requiring many hours of computer simulation time, rendering them of little use for real time applications, optimization, uncertainty quantification, sensitivity analysis, etc., all of which are crucial design tasks. Consequently, reducing the complexity of such models to perform simulations within an acceptable amount of time, with limited storage capacity, and with a reliable outcome is much needed.

As an example of how complex high-dimensional models arise in aerospace engineer-

ing, consider the following Navier-Stokes equation given in vector form as

$$\frac{\mathrm{D}U}{\mathrm{D}t} = \nu\nabla^2 U - \nabla p, \tag{2.5}$$

where ν represents the kinematic viscosity, $X = (x, y, z)^\top$ represents the spatial coordinate variables in vector form over the domain $\Omega \subset \mathbb{R}^3$, $U = (u(X,t), v(X,t), w(X,t))^\top$ represents the velocity vector field, $p = p(X,t)$ is the pressure scalar field, and $\frac{\mathrm{D}(\cdot)}{\mathrm{D}t}$ represents the material derivative given by $\frac{\partial(\cdot)}{\partial t} + U \cdot \nabla(\cdot)$. Expanding and rearranging the terms in equation 2.5 yields the following

$$\frac{\partial U}{\partial t} = \nu\nabla^2 U - (U \cdot \nabla) U - \nabla p. \tag{2.6}$$

It is clear that the local time derivative of the velocity field, $\frac{\partial U}{\partial t}$, is a nonlinear function of U, the first and second order spatial derivatives U_X, U_{XX}, and various other parameters. This is expressed in terms of the nonlinear function $N(\cdot)$ as

$$\frac{\partial U}{\partial t} = N(U) - \nabla p. \tag{2.7}$$

where $N(U) = -(U \cdot \nabla)U + \nu\nabla^2 U$, and the non-linearity is due to the quadratic term $(U \cdot \nabla) U$. It should be noted that equations 2.6 and 2.7 might deceivingly appear to be ODEs since only the local time derivative appears on the left hand side. However, these equations remain PDEs and have to be integrated temporally and spatially to realize a solution. A solution to an n^{th} order ODE is typically expressed as a linear combination of n independent solutions with n undetermined constants, therefore, a solution to an n^{th} order ODEs is said to be n-dimensional. On the other hand, a solution to an n^{th} order PDE is expressed as a linear combination of n independent solutions with n undetermined functions. Since those functions are infinite dimensional, solutions to PDEs are infinite dimensional. Many times, analytically integrating $\frac{\partial U}{\partial t}$ is impossible, leading engineers

down the path of numerical approximation.

Numerically approximating $\frac{\partial U}{\partial t}$ means that the solution $U(X,t)$, a continuous infinite dimensional variable, is to be discretized with respect to its spatial coordinates, X. This results in a finite dimensional representation of U with dimension $n \times 3$ where $n \in \mathbb{N}$ represents the number of grid points along each spatial axis, respectively, and 3 represents the number of flow field variables u, v, and w. The spatial discretization of $U(X,t)$ and $p(X,t)$ give rise to the spatially discretized velocity field $u^{\text{sd}}(t) \in \mathbb{R}^{n\times 3}$, $\forall t \in \mathbb{R}^+$ and the spatially discretized scalar pressure field $p^{\text{sd}}(t) \in \mathbb{R}^n$, which are equal to $U(X,t)$ and $p(X,t)$ respectively at the points of discretization, $\{X_i = (x_i, y_i, z_i)^\top\}_{i=1}^n$, at time t. Namely,

$$\begin{aligned} u_i^{\text{sd}}(t) &= U\left(X_i, t\right), \\ p_i^{\text{sd}}(t) &= p\left(X_i, t\right), \quad i = 1, \ldots, n. \end{aligned} \tag{2.8}$$

The choice of n is arbitrary and depends on the coarseness of the grid, with larger values of n resulting in finer grids.

After the grid is spatially discretized, the next step is to utilize an approximation scheme to approximate the spatial partial derivatives. Note that, at this stage, no discretization is done with respect to time. Some of the most common approximation schemes used are:

- Finite Difference (FD) approximation
- Finite Volume (FV) approximation
- Finite Element (FE) approximation

To illustrate the use of a first order central FD approximation scheme, an algebraic expression for the spatial derivatives is obtained. For example, consider the difference equation of u_i^{sd} along the x-axis given by

$$\begin{aligned} \frac{\partial U(X_i,t)}{\partial x} &\approx \frac{u_{i+1}^{\text{sd}} - u_{i-1}^{\text{sd}}}{2\Delta x}, \\ \frac{\partial^2 U(X_i,t)}{\partial x^2} &\approx \frac{u_{i+1}^{\text{sd}} - 2u_i^{\text{sd}} + u_{i-1}^{\text{sd}}}{\Delta x^2}. \end{aligned} \tag{2.9}$$

Using the same or a similar discretization scheme, an algebraic expression that approximates the derivatives in the direction of the remaining spatial variables and their higher order derivatives is obtained. Substituting the difference expressions for the spatial derivatives, like equations 2.9, with their respective approximations in equation 2.6, yields a large, but finite set of ordinary differential equations. This simplifies the partial time derivative in equation 2.6 to an ordinary time derivative given by

$$\frac{\mathrm{d}u^{\mathrm{sd}}}{\mathrm{d}t} = L\left(u^{\mathrm{sd}}\right) + Q\left(u^{\mathrm{sd}}, u^{\mathrm{sd}}\right) + H\left(p^{\mathrm{sd}}\right), \tag{2.10}$$

where $H : \mathbb{R}^n \to \mathbb{R}^{(n\times 3)}$, $L : \mathbb{R}^{n\times 3} \to \mathbb{R}^{n\times 3}$ are linear functions, and $Q : \mathbb{R}^{n\times 3} \times \mathbb{R}^{n\times 3} \to \mathbb{R}^{n\times 3}$ is bilinear. In order to remove an extra constant term, the discrete variables are offset by the steady state solution to the Navier-Stokes equation such that $u^{\mathrm{sd}} = 0$ is the steady state solution [30].

The system of ODEs given in equation 2.10 is numerically integrated to realize a numerical solution at different time steps. In the reduced-order modeling community, a solution at every time step is known as a snapshot [3]. Unfortunately, direct numerical integration of equation 2.10 might not always be possible. In CFD applications, the number of grid points n can be in the order of millions, which causes u^{sd} to have a very high-dimensionality rendering any numerical integration method impractical even on the best computational resources available. It is these high-dimensional systems that need to be reduced while preserving the system's behavior.

To summarize, performing a finite discretization of the spatial variables and substituting an approximate expression for the spatial derivatives reduce partial differential equations, like equation 2.6, to a large system of coupled ordinary differential equations. Model-order reduction aims to reduce the complexity of such high-dimensional ODEs to a lower order one while retaining as much of the input-output relationship and system's behavior as possible. In the next section, a formal definition of model-order reduction is introduced.

2.2.2 Formal Definition

Formally, model-order reduction is stated as follows:

Given a dynamical system G

$$G : \begin{cases} \dot{x}(t) = f(x(t), u(t)), \\ y(t) = g(x(t), u(t)), \end{cases} \tag{2.11}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, and $y \in \mathbb{R}^q$. Find the dynamical system G_r

$$G_\mathrm{r} : \begin{cases} \dot{a}(t) = f_\mathrm{r}(a(t), u(t)), \\ y_\mathrm{r}(t) = g_\mathrm{r}(a(t), u(t)), \end{cases} \tag{2.12}$$

such that $a \in \mathbb{R}^m$ and $m \ll n$, that also satisfies the following

$$\underset{G_\mathrm{r}}{\text{minimize}} \quad J\big(y\,(t)\,, y_\mathrm{r}\,(t)\,\big) \tag{2.13}$$

where $J(\cdot)$ is an error metric. Many reduced-order modeling techniques add further complexity to the problem by imposing constraints on G_r. These constraints ensure that system properties that exist in G (such as stability, passivity, etc.) also exist in G_r. Two classical and widely used error metrics are:

1. The relative mismatch error based on the H_2 norm defined as

$$J_{H_2} = \max_{t \geq 0} |y(t) - y_\mathrm{r}(t)| = \|G - G_r\|_{H_2} \|u\|_{L_2}. \tag{2.14}$$

2. The relative worst case error metric based on the H_∞ norm defined as

$$J_{H_\infty} = \sup_{\omega \in \mathbb{R}} \sigma_{\max}\big(G(jw) - G_\mathrm{r}(jw)\big) = \|G - G_\mathrm{r}\|_{H_\infty}. \tag{2.15}$$

In the case of linear time invariant dynamical systems, simplifications can be made, and G is given by

$$G : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t) + Du(t), \end{cases} \qquad \text{or} \qquad G(s) = C(sI - A)^{-1}B + D, \tag{2.16}$$

and the reduced-order model G_{r} is given as

$$G_{\mathrm{r}} : \begin{cases} \dot{a}(t) = A_{\mathrm{r}}a(t) + B_{\mathrm{r}}u(t), \\ y_{\mathrm{r}}(t) = C_{\mathrm{r}}a(t) + D_{\mathrm{r}}u(t), \end{cases} \qquad \text{or} \qquad G_r(s) = C_r(sI - A_r)B_r + D_r. \tag{2.17}$$

In the relative worst case error metric, merely evaluating the error metric is complex and challenging [57]. Gugercin et al. [58], demonstrate how to solve the model-order reduction problem for linear dynamical systems using the relative mismatch error by finding an approximate solution. An approximate solution needs to be found because finding an exact solution is infeasible for a variety of reasons, even for the linear case. According to Gugercin et al. [58], finding a global minimizer of $\|G-G_{\mathrm{r}}\|_{H_2}$ is difficult, and requires solving dense matrix operations that are computationally expensive, e.g., solution of a sequence of Lyapunov equations. Hence, iterative methods are used to find a suitable approximation G_{r}. In some situations, the dynamics of the full order model G is not available, only the inputs $u(t)$ and the outputs $y(t)$ are available and a simple model G_{r}, that produces $y_{\mathrm{r}}(t)$ which approximates $y(t)$, is desired. In such scenarios, system identification algorithms are helpful [36]. The field of model-order reduction is rich with algorithms suitable for different scenarios. Every algorithm is effective for different circumstances with different properties that an engineer might find appealing for his application. The purpose here is not to recount all the methods of model-order reduction, but to give a generalized theme of how model-order reduction works, and to focus on a particular method known as the proper orthogonal decomposition. For a detailed treatment of model-order reduction, the reader

should refer to the following books [25, 26, 28].

Typically, model-order reduction procedures follow a two-step approach. In the first step, a set of basis functions, sometimes known as modes, is computed. In the second step, a reduced-order model is constructed using the basis functions computed from step one. In the fluid flow model-order reduction community, the POD method is used to generate the basis functions. If the full-order model equations are available, a common procedure for step two is to project those equations onto the POD basis functions. This broad class of methods are known as projection-based model-order reduction methods [27]. Projection-based model-order reduction has shown great success in describing flow mechanics and its techniques generalize to nonlinear systems [30]. It is therefore discussed in more detail in the next section.

2.2.3 Projection-Based Model-Order Reduction

Model reduction via projection is one of the most ubiquitous techniques for model reduction specifically in flow mechanics. A formulation of the model reduction via projection is given by de Villemagne and Skelton in reference [59] and a survey is given by Benner et al. [27]. Given a nonlinear dynamical system, equation 2.11, the system of equations with full observation of the states are

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}x(t) &= f\big(x(t), u(t)\big), \\ y(t) &= Ix(t). \end{aligned} \tag{2.18}$$

Define S to be the subspace spanned by m unit vectors in the state space, $\phi_i \in \mathbb{R}^n, i = 1, \ldots, m$. The state vector $x(t)$ and its derivative $\dot{x}(t)$ are projected onto the subspace S

using the projection operation P_Φ as follows

$$\begin{aligned} P_\Phi\big(x(t)\big) &= \phi_1\phi_1^\top x(t) + \cdots + \phi_m\phi_m^\top x(t), \\ &= \underbrace{\begin{bmatrix} | & & | \\ \phi_1 & \dots & \phi_m \\ | & & | \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} — & \phi_1^\top x(t) & — \\ & \vdots & \\ — & \phi_m^\top x(t) & — \end{bmatrix}}_{a(t)}, \\ &= \Phi\Phi^\top x(t), \\ &= \Phi a(t), \\ &= \tilde{x}(t), \end{aligned} \tag{2.19}$$

where the vector $a(t) \in \mathbb{R}^m$ is referred to as the reduced state vector and $\tilde{x}(t) \in \mathbb{R}^n$ is the approximate state vector. Essentially the vector $a(t)$ is a set of coordinates that represent the approximate state vector $\tilde{x}(t)$ using the basis vectors $\{\phi_i\}_{i=1}^m$. The same projection process is applied to the state velocity vector $\dot{x}(t)$

$$\begin{aligned} P_\Phi\left(\frac{\mathrm{d}}{\mathrm{d}t}x(t)\right) &= \Phi\Phi^\top \frac{\mathrm{d}}{\mathrm{d}t}x(t), \\ &= \Phi\frac{\mathrm{d}}{\mathrm{d}t}\left(\Phi^\top x(t)\right), \\ &= \Phi\frac{\mathrm{d}}{\mathrm{d}t}a(t), \\ &= \Phi\dot{a}(t), \\ &= \dot{\tilde{x}}(t). \end{aligned} \tag{2.20}$$

Because $\tilde{x}(t)$ and $\dot{\tilde{x}}(t)$ are orthogonal projections of $x(t)$ and $\dot{x}(t)$, respectively, onto the subspace S, then $(x - \tilde{x}) \in S^\perp$ and $(\dot{x} - \dot{\tilde{x}}) \in S^\perp$. Therefore, $\tilde{x}(t)$ and $\dot{\tilde{x}}$ are the best

approximates of $x(t)$ and $\dot{x}(t)$ in the subspace S, which is stated as follows

$$\tilde{x}(t) = \arg\min_{v \in S} \|v - x(t)\|^2, \tag{2.21}$$

$$\dot{\tilde{x}}(t) = \arg\min_{v \in S} \|v - \dot{x}(t)\|^2. \tag{2.22}$$

Now that the reduced state vector $a(t)$ is defined as coordinates of the projection of the original state vector onto the subspace S spanned by the choice of basis vectors Φ, the next step is to find a set of equations that can be used to model the evolution of the reduced states such that $x(t) \approx \tilde{x}(t) = \Phi a(t), \forall t$. In other words, an equation of motion that dictates the evolution of the reduced state vector $a(t)$ is sought after. Note that the derivative and the projection operations commute, therefore, derivative of the projected state vector, is equal to the projection of the state derivative, i.e.

$$\frac{\mathrm{d}}{\mathrm{d}t}\Big(P_\Phi\big(x\,(t)\,\big)\Big) = P_\Phi\left(\frac{\mathrm{d}}{\mathrm{d}t}x(t)\right), \tag{2.23}$$

$$\dot{\tilde{x}} = P_\Phi\Big(f\big(x(t)\big)\Big). \tag{2.24}$$

It is tempting to assume that $\dot{\tilde{x}} = f(\tilde{x}, u(t))$, however, generally speaking

$$\dot{\tilde{x}}(t) \neq f\Big(\tilde{x}(t), u(t)\Big). \tag{2.25}$$

Although the function $f(\cdot)$ maps the state vector $x(t)$ to its derivative $\dot{x}(t)$, that does not imply that it maps the approximated state vector $\tilde{x}(t)$ to its derivative $\dot{\tilde{x}}(t)$. This is due to the fact that f and P_Φ do not necessarily commute, i.e.

$$\dot{\tilde{x}} = P_\Phi\Big(f\big(x(t)\big)\Big) \neq f\Big(P_\Phi\big(x(t)\big)\Big). \tag{2.26}$$

By definition $\tilde{x}$ and $\dot{\tilde{x}}$ are projections, and therefore constrained to be in the subspace S, however $f(\tilde{x})$ does not have the same restrictions.

It is desirable to find a reduced-order model (a model that dictates the evolution of the approximated states in terms of the reduced coordinates) where the velocity residual $\varepsilon(t)$ is minimized. The velocity residual is defined as follows

$$\begin{aligned} \varepsilon(t) &= P_\Phi\Big(f\big(x(t)\big)\Big) - f\Big(P_\Phi\big(x(t)\big)\Big), \\ &= \dot{\tilde{x}}(t) - f(\tilde{x}(t)). \end{aligned} \tag{2.27}$$

Generating a reduced-order model that minimizes velocity residual $\varepsilon(t)$ by constraining the velocity residual to be orthogonal to the subspace S is known as Galerkin approximation [23]. By definition of orthogonality,

$$< \phi_i \, , \varepsilon >= 0, \quad i = 1, \ldots, m, \tag{2.28}$$

$$\iff \Phi^\top\Big(\dot{\tilde{x}}(t) - f\big(\tilde{x}(t)\big)\Big) = 0. \tag{2.29}$$

Hence, minimizing the residual error via the Galerkin approximation demands that

$$\begin{aligned} \Phi^\top\dot{\tilde{x}}(t) &= \Phi^\top f(\tilde{x}), \\ \Phi^\top\Phi\dot{a}(t) &= \Phi^\top f(\Phi a(t)). \end{aligned} \tag{2.30}$$

If the unit vectors ϕ_i were chosen to be orthogonal to each other, then $\Phi^\top\Phi = I$ and

$$\dot{a}(t) = \Phi^\top f\Big(\Phi a(t)\Big). \tag{2.31}$$

Therefore, the reduced-order model is optimal in the sense of minimizing the velocity residual $\varepsilon(t)$. For the sake of clarity, Carlberg et al. [60] equivalently phrase the result of equation 2.31 as follows: if $\Phi^\top\Phi = I$, then

$$\frac{\mathrm{d}}{\mathrm{d}t}a(t) = \arg\min_{v\in\mathbb{R}^m} \|\Phi v - f\big(\tilde{x}(t), u(t)\big)\|^2. \tag{2.32}$$

Orthogonal projection-based model-order reduction (also known as Galerkin projection) only requires an orthonormal set of basis vectors $\{\phi_i\}_{i=1}^m$. With an orthonormal basis vectors, the state vector $x(t)$ and the state function $f(\cdot)$ can both be projected onto a lower-dimensional subspace S to derive a reduced-order model in an optimal fashion.

In summary, one set of projection reduced-order models, Galerkin reduced-order models, are created using the same two step process described in section 2.2.2. In the first step, a set of m orthogonal unit vectors $\{\phi_i\}_{i=1}^m$ are chosen. In the second step, the full-order model is reduced by projecting its equations of motion onto a lower dimensional subspace spanned by the basis vectors from step one, resulting in a reduced-order model

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}a(t) &= \Phi^\top f\left(\Phi a(t), u(t)\right), \quad a(0) = \Phi^\top x(0), \\ y_\mathrm{r}(t) &= \Phi a(t). \end{aligned} \tag{2.33}$$

These reduced equations are now of order m as opposed to the full-order model which has an order of n. The reduced-order state equation is integrated and the original model is approximated via

$$y(t) = x(t) \approx y_\mathrm{r}(t) = \tilde{x}(t) = \sum_{i=1}^{m} \phi_i a_i(t) = \Phi a(t) \tag{2.34}$$

The procedure described outlines how to construct an m-dimensional projection-based reduced-order model using any set of orthogonal unit vectors $\{\phi_i\}_{i=1}^m$. In order to construct an effective reduced-order model, special attention should be given to the choice of the basis vectors $\{\phi_i\}_{i=1}^m$. It is desirable to generate a reduced-order model that accurately approximates the full-order model by capturing the essential dynamics in the full-order model with as few basis functions as possible, i.e. $m \ll n$. In the next section, the proper orthogonal decomposition method for finding the optimal basis functions is described. The proper orthogonal decomposition is chosen as it is the most ubiquitous and promising method used in computing the basis functions.

2.3 Proper Orthogonal Decomposition

The proper orthogonal decomposition (POD) has enjoyed great success in many fields of science and engineering, such as machine learning, probability and statistics, signal processing, etc. In the literature, the POD is also known as the Karhunen-Loève decomposition (K-L decomposition), principle components analysis (PCA), and empirical eigenfunction decomposition.

The POD was first introduced to the fluid flow community by Lumley [1] in 1967 to better understand and analyze turbulent flow. In his approach to characterize turbulent flow, Lumley derived the POD method in an attempt to provide a mathematical description of repeating patterns in the flow, which he called coherent structures. The success of the POD in decomposing complex flows has made the method a focal point of investigation, and a central theme for the modal analysis of fluid flows. Moreover, the coherent structures computed via the POD method can be used as basis function which enables the dynamical modeling of fluid systems [23]. The success of the POD has resulted in many variations, adaptations, and applications of the POD method [30, 4]. In the next section, the POD is investigated in greater depth. A full treatment of the topic is available by Holmes, Lumley and Berkooz [61].

2.3.1 Turbulence and Coherent Structures

Experimental and theoretical advancements in fluid dynamics suggest a new approach to turbulent flow. A physical phenomenon is said to be random when its future behavior cannot be predicted within reasonable experimental error. While at first sight turbulent flow seems to be random and unpredictable, recent advancements have shown underlying patterns that can be exploited in predicting turbulent flow. Turbulent flow is governed by dissipative partial differential equations, the Navier-Stokes equations (equations 2.3). Dissipative chaos theory has shown that the long term behavior of such PDEs resides on a

finite and sometimes low dimensional manifold known as strange attractors [61]. Experimentally, it has also been shown that these attractors exist. Lumley termed these attractors as coherent structures. Coherent mean persistent and reappearing, and structures mean flow patterns. Lumley [1] in 1967, was first to suggest a mathematical procedure for extracting and describing coherent structures that can be used for further analysis. The procedure consists of taking the proper orthogonal decomposition of the two point spatial velocity correlation matrix, even though the mathematical derivation of these structures does not necessarily coincide with the experimental ones [5]. For an in depth treatment of the subject, refer to references [61, 62].

To properly establish the derivation of the proper orthogonal decomposition for extracting coherent structures, a few assumptions have to be made. For incompressible flows the following is an expression of the Navier-Stokes equation

$$\frac{\mathrm{D}\,U}{\mathrm{D}\,t} + \nabla p = \nu \nabla^2 U, \tag{2.35}$$

$$\nabla \cdot U = 0, \tag{2.36}$$

with the appropriate boundary conditions. The flows are assumed to be turbulent but time stationary and ergodic. Time stationary implies that any statistic computed over the ensembles remains constant with time, i.e. statistical properties do not change with time. For example, the ensemble average of the speed in the x-direction $\mathbb{E}[u(X,t)]$, at a certain location X and time t, is given by the average of the observations $u_i(X,t)$ where the index i represents a different ensemble (a different realization of a stochastic process)

$$\begin{aligned} \mathbb{E}[u(X,t)] &= \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} u_i(X,t), \\ &= C, \quad \forall t \in \mathbb{R}^+. \end{aligned} \tag{2.37}$$

Assuming ergodicity allows for the computation of statistics using observations from one

ensemble collected over time instead of commuting statistics over many ensembles at a fixed time. Proceeding with the same example of the average speed in the x-direction and applying the principle of ergodicity

$$\begin{aligned}\mathbb{E}[u(X,t)] &= \lim_{N\to\infty}\sum_{i=1}^{N} u_i(X,t), \\ &= \lim_{T\to\infty}\int_0^T u(X,\tau)\,\mathrm{d}\tau, \quad \forall t \in \mathbb{R}^+.\end{aligned} \tag{2.38}$$

This enables the computation of ensemble averages such that the flow can then be separated into a sum of its mean and fluctuation around the mean. With a slight abuse of notation, the fluctuations are denoted by $v(\cdot,\cdot)$,

$$v(X,t) = U(X,t) - \mathbb{E}\left[U(X,t)\right]. \tag{2.39}$$

Hence $v(X,t)$ is now a stochastic process with zero mean. Let $X_1, X_2 \in \mathbb{R}^3$ represent the Cartesian coordinates of two points in the domain of a fluid, the two point spatial correlation function $R(\cdot,\cdot)$ is defined as

$$R(X_1,X_2) = \mathbb{E}\left[v(X_1,t)v^\star(X_2,t)\right]. \tag{2.40}$$

Since $v(\cdot,\cdot) \in \mathbb{R}^3$, the complex conjugate transpose is reduced to a transpose, and using the property of ergodicity, the two point spatial correlation function is then re-expressed as

$$R(X_1,X_2) = \frac{1}{T}\int_0^T v(X_1,\tau)v^\top(X_2,\tau)\,\mathrm{d}\tau. \tag{2.41}$$

The fluid velocity fluctuation variable $v(\cdot,\cdot)$, can also be expressed as the superposition of spatial modes multiplied by time coefficients as follows

$$v(X,t) = \sum_{i=1}^{\infty}\phi_i(X)a_i(t). \tag{2.42}$$

Any set of admissible spatial functions can be used. Admissible is used here to imply orthonormal functions that meet the boundary conditions. Hence, every $\phi_i(X)$ that satisfies the boundary conditions of the original problem.

In order to create a reduced representation of the flow, the infinite sum is truncated to include only the first m terms. This results in an approximate representation v_m of order m given by

$$v_m(X,t) = \sum_{i=1}^{m} \phi_i(X) a_i(t). \tag{2.43}$$

This poses an interesting question, what m spatial modes give rise to the best approximation of the flow? Formally, this is stated as the following optimization problem

$$\begin{aligned} &\underset{\{\phi_i(X)\}_{i=1}^m}{\text{minimize}} \quad \epsilon_m = \int_0^T \left\| v\left(X,t\right) - \sum_{i=1}^{m} \left\langle v\left(X,t\right) \, , \phi_i\left(X\right)\right\rangle_X \phi_i\left(X\right) \right\|_X^2 \mathrm{d}t, \\ &\text{subject to} \quad \phi_i(X)^\top \phi_j\left(X\right) = \delta_{ij}, \quad 1 \leq i,j \leq m. \end{aligned} \tag{2.44}$$

An optimal solution $\{\phi_i\}_{i=1}^m$ to the minimization problem is called a POD basis of rank m. With some algebraic manipulation of the integrand, the minimization problem (equation 2.44), can be restated as a maximization problem. The integrand in equation 2.44 is manipulated as follows

$$\begin{aligned} \left\| v\left(X,t\right) - \sum_{i=1}^{m} \left\langle v\left(X,t\right) \, , \phi_i\right\rangle \phi_i \right\|_X^2 &= \left\langle v\left(X,t\right) - \sum_{i=1}^{m} \left\langle v\left(X,t\right), \phi_i\right\rangle_X \phi_i \, , \right. \\ &\qquad \left. v\left(X,t\right) - \sum_{i=1}^{m} \left\langle v\left(X,t\right), \phi_j\right\rangle_X \phi_j \right\rangle_X , \\ &= \left\| v\left(X,t\right) \right\|_X^2 - 2 \sum_{i=1}^{m} \left\langle v(X,t) \, , \phi_i \right\rangle_X^2 \\ &\quad + \sum_{i=1}^{m} \sum_{j=1}^{m} \left\langle v(X,t) \, , \phi_i \right\rangle_X \left\langle v(X,t) \, , \phi_j \right\rangle_X \phi_i^\top \phi_j, \\ &= \left\| v\left(X,t\right) \right\|_X^2 - \sum_{i=1}^{m} \left\langle v(X,t) \, , \phi_i \right\rangle_X^2 . \end{aligned} \tag{2.45}$$

Since $\|v\left(X,t\right)\|_X^2$ is a constant and does not change for a given problem, minimizing the previous expression is equivalent to maximizing the sum of the projections squared. Hence, analogous to the minimization problem is maximizing the integral of the sum of the length squared of the projections of $v(X,t)$ onto $\phi(X)$, the subspace spanned by $\phi_1(X),\ldots,\phi_m(X)$. The property of additivity of the inner product yields a more compact representation of the minimization problem stated as a maximization problem

$$\begin{aligned} &\underset{\phi(X)}{\text{maximize}} \quad \int_0^T \left\langle v\left(X,t\right)\,,\phi\left(X\right)\right\rangle_X^2 \,\mathrm{d}t, \\ &\text{subject to} \quad \left\|\phi\left(X\right)\right\|_X^2 = 1. \end{aligned} \tag{2.46}$$

In order to solve the maximization problem, the following functional is defined

$$\begin{aligned} J\left[\phi\left(X\right)\right] &= \int_0^T \left\langle v\left(X,t\right)\,,\phi\left(X\right)\right\rangle_X^2 \,\mathrm{d}t - \lambda\left(\left\|\phi\left(X\right)\right\|^2 - 1\right), \\ &= \int_0^T \left\langle v\left(X,t\right)\,,\phi\left(X\right)\right\rangle_X \left\langle v\left(X,t\right)\,,\phi\left(X\right)\right\rangle_X \,\mathrm{d}t - \lambda\left(\left\langle\phi\left(X\right)\,,\phi\left(X\right)\right\rangle - 1\right). \end{aligned} \tag{2.47}$$

A necessary condition for ϕ to be an extremal of J is that the functional derivative of J must equal zero for all possible variations $\phi+\delta\psi, \delta\in\mathbb{R}$. Mathematically, this is expressed as

$$\left.\frac{\mathrm{d}}{\mathrm{d}\delta}\right|_{\delta=0} J\left[\phi\left(X\right)+\delta\psi\left(X\right)\right] = 0, \quad \forall\psi\in H. \tag{2.48}$$

The expression is simplified as follows

$$\begin{aligned} \left.\frac{\mathrm{d}}{\mathrm{d}\delta}\right|_{\delta=0} J\left[\phi+\delta\psi\right] &= \frac{\mathrm{d}}{\mathrm{d}\delta}\left[\int_0^T \left\langle v\,,\phi+\delta\psi\right\rangle_X \left\langle v\,,\phi+\delta\psi\right\rangle_X \mathrm{d}t\right. \\ &\qquad \left. - \lambda\left(\left\langle\phi+\delta\psi\,,\phi+\delta\psi\right\rangle_X - 1\right)\right]\Bigg|_{\delta=0}, \qquad \forall\psi\in H, \\ &= 2\int_0^T \left\langle v\,,\psi\right\rangle_X \left\langle v\,,\phi\right\rangle_X \,\mathrm{d}t - 2\lambda\left\langle\phi\,,\psi\right\rangle_X, \qquad \forall\psi\in H, \\ &= 2\int_0^T \left\langle v\left\langle v\,,\phi\right\rangle_{X'}\,,\psi\right\rangle_X \,\mathrm{d}t - 2\lambda\left\langle\phi\,,\psi\right\rangle_X, \qquad \forall\psi\in H. \end{aligned} \tag{2.49}$$

Note that the inner product $< \cdot, \cdot >_X$ is an integral with respect to the spatial coordinates and the order of spatial integral can be swapped with time integral. Setting the first variation to zero and noting that the inner product is commutative for real numbers leads to

$$\begin{aligned} 0 &= \left\langle \int_0^T v(X,t)\left\langle v(X',t)\,,\phi(X')\right\rangle_{X'} \mathrm{d}t\,,\psi(X)\right\rangle_X - \lambda\left\langle \phi(X)\,,\psi(X)\right\rangle_X, \qquad \forall\psi\in H,\\ &= \left\langle \left\langle \int_0^T v(X,t)v^\top(X',t)\,\mathrm{d}t\,,\phi(X')\right\rangle_{X'},\psi(X)\right\rangle_X - \lambda\left\langle \phi(X)\,,\psi(X)\right\rangle_X, \quad \forall\psi\in H,\\ &= \left\langle \left\langle \int_0^T v(X,t)v^\top(X',t)\,\mathrm{d}t\,,\phi(X')\right\rangle_{X'} - \lambda\phi(X)\,,\psi(X)\right\rangle_X, \qquad \forall\psi\in H. \end{aligned} \tag{2.50}$$

Since the previous expression shows that the inner product is always zero $\forall\psi \neq 0 \in H$, it must be that

$$\int_\Omega \int_0^T v(X,t)v^\top(X',t)\,\mathrm{d}t\,\phi(X')\mathrm{d}X' - \lambda\phi(X) = 0. \tag{2.51}$$

This reveals the integral eigenvalue-eigenvector problem that optimally solves the maximization problem and is an extremal of J. This is given by

$$\int_\Omega R(X,X')\phi(X')\,\mathrm{d}X' = \lambda\phi(X). \tag{2.52}$$

The eigenvectors, $\phi(X)$, is what Lumley referred to as the coherent structure. The eigenvalue problem in equation 2.52 is known as the Fredholm integral equation [63]. The interested reader should refer to Fredholm theory for an in-depth treatment of the topic. In the next section, the computational methods that implement the proper orthogonal decomposition method are described.

2.3.2 Computational Methods

The first implementation, first popularized by Lumley [1] in the fluid flow community, is known as the direct method. It has been coined as the direct method since the implemen-

tation directly perform the decomposition suggested by Lumley on the discretized version of equation 2.52. The second implementation of the proper orthogonal decomposition, is known as the method of snapshots, first devised by Sirovich [3] and has gained large success and popularity. The snapshot method follows an indirect approach in computing a discretized decomposition of equation 2.52.

Both implementations yield the same results, albeit the snapshot method is computationally tractable and the direct method is not. The snapshot implementation of the POD method has largely contributed to the popularity of the POD method, as it offers huge computational time and memory savings. The snapshot implementation allows for the POD method to be scalable to problems with large data sets such as the ones encountered in modern day CFD problems.

In numerical simulations, the temporal and spatial variables are discretized. Hence, the integral in the eigenvalue problem given by equation 2.52 is no longer applicable. As a result, the discretized version of the two spatial correlation function is used instead. The flow field variables of interest are arranged column wise, where every column is the numerical data obtained at a snapshot of time (specific time instant) from either measurements taken in an experiment or numerical data computed via a CFD simulation. The following data matrices are defined

$$U^{\mathrm{d}} = \begin{bmatrix} u_1^1 & u_1^2 & \dots & u_1^N \\ \vdots & \vdots & \dots & \vdots \\ u_n^1 & u_n^2 & \dots & u_n^N \end{bmatrix}, V^{\mathrm{d}} = \begin{bmatrix} v_1^1 & v_1^2 & \dots & v_1^N \\ \vdots & \vdots & \dots & \vdots \\ v_n^1 & v_n^2 & \dots & v_n^N \end{bmatrix}, W^{\mathrm{d}} = \begin{bmatrix} w_1^1 & w_1^2 & \dots & w_1^N \\ \vdots & \vdots & \dots & \vdots \\ w_n^1 & w_n^2 & \dots & w_n^N \end{bmatrix}, \tag{2.53}$$

where $U^d, V^d, W^d \in \mathbb{R}^{n \times N}$, n represents the number of grid points, N represents the number of snapshots (time instances where data is collected), the superscript indicates the snapshot instance, and the subscript indicates the grid number. The data matrices U^d, V^d and W^d are then concatenated (by vertically stacking them on top of each other) into a big

data matrix $u^{\mathrm{d}} \in \mathbb{R}^{(n\times 3)\times N}$ as follows

$$u^{\mathrm{d}} = \begin{bmatrix} U^{\mathrm{d}} \\ V^{\mathrm{d}} \\ W^{\mathrm{d}} \end{bmatrix} = \begin{bmatrix} u_1^1 & u_1^2 & \dots & u_1^N \\ \vdots & \vdots & \dots & \vdots \\ u_n^1 & u_n^2 & \dots & u_n^N \\ v_1^1 & v_1^2 & \dots & v_1^N \\ \vdots & \vdots & \dots & \vdots \\ v_n^1 & v_n^2 & \dots & v_n^N \\ w_1^1 & w_1^2 & \dots & w_1^N \\ \vdots & \vdots & \dots & \vdots \\ w_n^1 & w_n^2 & \dots & w_n^N \end{bmatrix}. \tag{2.54}$$

The matrix u^{d} is known as the snapshot matrix in the literature, and its columns $\mathrm{col}_i(u^{\mathrm{d}})$ are known as the snapshots or strobes [3]. The norm of the data matrix u^{d} is given by

$$\|u^{\mathrm{d}}\|_{\mathrm{F}}^2 = \|U^{\mathrm{d}}\|_{\mathrm{F}}^2 + \|V^{\mathrm{d}}\|_{\mathrm{F}}^2 + \|W^{\mathrm{d}}\|_{\mathrm{F}}^2. \tag{2.55}$$

The Direct Method

In order to find the correlation of the flow field variables between any two grid points, the discrete spatial correlation matrix is constructed as follows

$$\begin{aligned} R^{\mathrm{d}} &= \frac{1}{N} u^{\mathrm{d}} {u^{\mathrm{d}}}^{\top}, \\ &= \frac{1}{N} \begin{bmatrix} U^{\mathrm{d}} \\ V^{\mathrm{d}} \\ W^{\mathrm{d}} \end{bmatrix} \begin{bmatrix} {U^{\mathrm{d}}}^{\top} & {V^{\mathrm{d}}}^{\top} & {W^{\mathrm{d}}}^{\top} \end{bmatrix}, \\ &= \frac{1}{N} \begin{bmatrix} U^{\mathrm{d}}{U^{\mathrm{d}}}^{\top} & U^{\mathrm{d}}{V^{\mathrm{d}}}^{\top} & U^{\mathrm{d}}{W^{\mathrm{d}}}^{\top} \\ V^{\mathrm{d}}{U^{\mathrm{d}}}^{\top} & V^{\mathrm{d}}{V^{\mathrm{d}}}^{\top} & V^{\mathrm{d}}{W^{\mathrm{d}}}^{\top} \\ W^{\mathrm{d}}{U^{\mathrm{d}}}^{\top} & W^{\mathrm{d}}{V^{\mathrm{d}}}^{\top} & W^{\mathrm{d}}{W^{\mathrm{d}}}^{\top} \end{bmatrix}. \end{aligned} \tag{2.56}$$

where $R^{\mathrm{d}} \in \mathbb{R}^{(n\times 3)\times(n\times 3)}$. The discretized eigenvalue problem becomes

$$R^{\mathrm{d}}\phi_i = \phi_i \lambda_i, \tag{2.57}$$

After the eigenvalues and eigenvectors of the matrix R^{d} are computed, the components of each eigenvector can be extracted as follows

$$\phi_i^u = \mathrm{row}_{j=1}^{n}\left(\phi_i\right), \tag{2.58}$$

$$\phi_i^v = \mathrm{row}_{j=n+1}^{n\times 2}\left(\phi_i\right), \tag{2.59}$$

$$\phi_i^w = \mathrm{row}_{j=n\times 2+1}^{n\times 3}\left(\phi_i\right). \tag{2.60}$$

The eigenvectors ϕ_i are known as the POD modes or coherent structures.

Since the size of matrix R^{d} can be tremendously large, the eigen decomposition of such a matrix can be an impossible task due to computational resource limitations. In order to circumvent this issue, the snapshot implementation was devised [3]. In the next section, the snapshot implementation of the POD method is outlined.

The Snapshot Method

As mentioned, the direct method is not computationally manageable to perform. This is because the number of grid points for most current engineering applications can be large, which renders the dimensions of matrix R^{d} to be even larger, therefore solving it on a computer is restricted due to memory and computational limitations. Sirovich [3] devised a computationally tractable version of the POD method that yields the same results and named it the snapshot method.

At the heart of the snapshot method is an algebraic trick, namely, the eigenvalue de-

composition of ${u^{\mathrm{d}}}^{\top}u^{\mathrm{d}}$ is solved instead of $u^{\mathrm{d}}{u^{\mathrm{d}}}^{\top}$. The matrix ${u^{\mathrm{d}}}^{\top}u^{\mathrm{d}}$ is given by

$$ {u^{\mathrm{d}}}^{\top}u^{\mathrm{d}} = \begin{bmatrix} {U^{\mathrm{d}}}^{\top} & {V^{\mathrm{d}}}^{\top} & {W^{\mathrm{d}}}^{\top} \end{bmatrix} \begin{bmatrix} U^{\mathrm{d}} \\ V^{\mathrm{d}} \\ W^{\mathrm{d}} \end{bmatrix} = {U^{\mathrm{d}}}^{\top}U^{\mathrm{d}} + {V^{\mathrm{d}}}^{\top}V^{\mathrm{d}} + {W^{\mathrm{d}}}^{\top}W^{\mathrm{d}}. \tag{2.61} $$

The eigenvalues and eigenvectors of matrix ${u^{\mathrm{d}}}^{\top}u^{\mathrm{d}}$ are computed by solving the following eigenvalue problem

$$ {u^{\mathrm{d}}}^{\top}u^{\mathrm{d}}\psi_i(X) = \lambda_i\psi_i(X). \tag{2.62} $$

In this case $({u^{\mathrm{d}}}^{\top}u^{\mathrm{d}}) \in \mathbb{R}^{N\times N}$, where N is number of snapshots, a much smaller number than n, the number of grid points. The eigenvectors that arise from equation 2.62 and the eigenvectors that arise from equation 2.57 are related via the following relationship

$$ \phi_i = u^{\mathrm{d}}\psi_i\frac{1}{\sqrt{\lambda_i}}. \tag{2.63} $$

which can be written in matrix form as

$$ \Phi = u^{\mathrm{d}}\Psi\Lambda^{-\frac{1}{2}}. \tag{2.64} $$

where $\Phi = [\phi_1, \ldots, \phi_m]$, $\Psi = [\psi_1, \ldots, \psi_m]$, and $\Lambda = \mathrm{diag}(\lambda_1 \ldots \lambda_N)$. Equations 2.63 and 2.64, can be easily derived by multiplying both sides of equation 2.62 by u^{d} and comparing the result to equation 2.57 as follows

$$ u^{\mathrm{d}}{u^{\mathrm{d}}}^{\top}u^{\mathrm{d}}\psi_i(X) = u^{\mathrm{d}}\lambda_i\psi_i(X), \tag{2.65} $$

$$ \iff \underbrace{\left(u^{\mathrm{d}}{u^{\mathrm{d}}}^{\top}\right)}_{R^{\mathrm{d}}}\underbrace{u^{\mathrm{d}}\psi_i(X)}_{\phi_i(X)} = \lambda_i\underbrace{u^{\mathrm{d}}\psi_i(X)}_{\phi_i(X)}. \tag{2.66} $$

Hence, given a numerical simulation of fluid flow with N snapshots, the matrix $S^{\mathrm{d}} \in$

$\mathbb{R}^{N\times N}$ is constructed as follows

$$S^{\mathrm{d}} = {u^{\mathrm{d}}}^{\top} u^{\mathrm{d}}.$$

Since in most numerical applications, the number of snapshots N is much smaller than the number of grid points n, ($N \ll n$), $S^{\mathrm{d}} \in \mathbb{R}^{N\times N}$ is much smaller than $\mathbb{R}^{\mathrm{d}} \in \mathbb{R}^{(n\times 3)\times(n\times 3)}$. The snapshot method relies on taking the eigenvalue decomposition of S^{d} and then transforming the result into the POD modes. In this manner, the POD modes can be computed without having to compute the matrix R^{d}.

A Note on the Singular Value Decomposition

It is well known from linear algebra that the eigenvalue decomposition (EVD) is intimately related to the singular value decomposition (SVD). Specifically speaking, given a matrix $A \in \mathbb{R}^{p\times q}$, the left singular vectors of A are the eigenvectors of $AA^{\top}$, and the right singular vectors of A, are the eigenvectors of $A^{\top}A$ [4]. The singular values of A are also the square root of the eigenvalues of $AA^{\top}$ and $A^{\top}A$. This means that the singular value decomposition can be utilized as a means of implementing the POD method. The singular value decomposition of matrices A and $A^{\top}$, are defined as the eigenvalue decomposition of the matrices $AA^{\top}$ and $A^{\top}A$,

$$\mathrm{SVD}(A) = \mathrm{EVD}(AA^{\top}), \tag{2.67}$$

$$\mathrm{SVD}(A^{\top}) = \mathrm{EVD}(A^{\top}A). \tag{2.68}$$

However, the numerical computation of the singular value decomposition is much more numerically stable than the eigenvalue decomposition, therefore, it has become the method of choice for performing the decompositions given in equation 2.57 and equation 2.62. Moreover, the singular value decomposition offers deep insights in matrix algebra which might aid in the understanding of coherent structures. Therefore, it is important to discuss the singular value decomposition and how it applies.

Typically, the singular value decomposition is viewed as a decomposition of a finite dimensional linear operator (matrix) into more elementary operations. The singular value decomposition states that every linear operator between two vector spaces can be expressed in terms of three more elementary operations, namely, a rotation first, scaling second, and then another rotation. For example, given two finite dimensional vector spaces $X \in R^N$ and $Y \in R^n$, let the linear mapping $A \in \mathbb{R}^{n \times N}$ be such that $A : X \to Y$. The singular value decomposition states that all linear mappings can be decomposed as follows

$$A = U\Sigma V^\top$$

where, $U \in \mathbb{R}^{n \times n}, V \in \mathbb{R}^{N \times N}$ are unitary and $\Sigma \in \mathbb{R}^{n \times N}$ is diagonal. This decomposition holds for all linear operators regardless of the dimension of either the domain X, or the target space Y.

However, using the singular value decomposition as an implementation of the POD implies taking the singular value decomposition of the matrix u^{d}. It is confusing to think of the matrix u^{d} as a linear operation between two vector spaces. What does it mean when the matrix u^{d} is constructed from CFD data arranged into columns? What linear operation does that matrix perform, and what does it mean to take the singular value decomposition of such a matrix?

In the mathematical perspective of interpreting matrices as linear operator between two vector spaces, little to no insight is added to the problem. However, the singular value decomposition of a matrix can be seen under another light. The singular value decomposition has other properties that relate to the columns of a matrix. One geometric interpretation is that the singular vectors of u^{d} are the principal axes that minimizes the variance of the data points. A second geometric interpretation is that the singular value decomposition attempts to find a subspace, that maximizes the norm of vectors projected onto that subspace, the vectors in this case are the snapshot, i.e. the columns of the matrix u^{d}.

Consider the following optimization problem, it will be shown that the singular value decomposition provides the solution. Let $a_1, \ldots, a_N \in \mathbb{R}^n$, let U be the subspace spanned by the unit vectors $u_1, \ldots, u_m \in \mathbb{R}^n$,

$$\begin{aligned} \underset{\{u_i\}_{i=1}^m}{\text{maximize}} \quad & \sum_{i=1}^{N} \|P_U(a_i)\| \\ \text{subject to} \quad & \|u_i\| = 1, \quad \forall i = 1, \ldots, m. \end{aligned} \tag{2.69}$$

where $P_U(\cdot)$ is the projection of a vector onto the subspace U. With slight abuse of notation, let the matrix $U = [u_1, \ldots, u_m]$. The projection of an arbitrary vector a_i, on the subspace spanned by the columns of matrix U is given by $P_U(a_i) = UU^\top a_i$. Hence, the following sums are equivalent $\sum_{i=1}^{N} \|P_U(a_i)\| = \sum_{i=1}^{N} \left\|UU^\top a_i\right\|$. Note that squaring the norms will simplify the math without changing the solution, hence an equivalent problem is trying to maximize $\sum_{i=1}^{N} \left\|UU^\top a_i\right\|^2$. By arranging the vectors a_i into the columns of a matrix $A = [a_1, \ldots, a_N]$, the summation is automatically accounted for as a matrix operation since

$$\sum_{i=1}^{m} \|UU^\top \text{col}_i(A)\|^2 = \sum_{i=1}^{m} \|\text{col}_i(UU^\top A)\|^2 = \|UU^\top A\|^2. \tag{2.70}$$

Moreover, imposing the constraint of $\|U\| = 1$ makes it is enough to maximize the dot product $U^\top A$ instead of $UU^\top A$, which results in a more compact representation of the maximization problem. Hence, the following is an equivalent maximization problem to that of equation 2.69

$$\begin{aligned} \underset{U \in \mathbb{R}^{n \times m}}{\text{maximize}} \quad & \left\|U^\top A\right\|^2 \\ \text{subject to} \quad & \|U\|^2 = 1, \quad \forall i = 1, \ldots, m. \end{aligned} \tag{2.71}$$

In order to solve this optimization problem, the Lagrangian is constructed and its partial derivative are set to zero as follows (note that the norm of a matrix can be expressed in terms

of the trace)

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial U} &= \frac{\partial}{\partial U}\Bigg(\text{trace}\left(U^\top A\left(U^\top A\right)^\top\right) - \Lambda\left(\text{trace}\left(UU^\top\right) - 1\right)\Bigg) \\
&= \frac{\partial}{\partial U}\Bigg(\text{trace}\left(U^\top AA^\top U\right) - \Lambda\left(\text{trace}\left(UU^\top\right) - 1\right)\Bigg), \\
&= \frac{\partial}{\partial U}\Bigg(\text{trace}\left(AA^\top UU^\top\right) - \Lambda\left(\text{trace}\left(UU^\top\right) - 1\right)\Bigg), \\
&= \frac{\partial}{\partial U}\Bigg(\text{trace}\left(AA^\top UU^\top\right)\Bigg) - \Lambda\frac{\partial}{\partial U}\Bigg(\text{trace}\left(UU^\top\right)\Bigg), \\
&= \Bigg(AA^\top + \left(AA^\top\right)^\top\Bigg)U - \Lambda\Bigg(I + I^\top\Bigg)U, \\
&= 2\left(AA^\top - \Lambda I\right)U, \\
&= 0.
\end{aligned}$$

Hence, the matrix U, that sets the previous partial derivative to zero satisfies the following relationship

$$AA^\top U = \Lambda U.$$

Therefore, the columns of U are the eigenvectors of $AA^\top$. Using simple linear algebra arguments, it can also be shown that the error of the minimization problem is given by

$$\sum_{i=m+1}^{N} \lambda_i(A) = \underset{U\in\mathbb{R}^{n\times m}}{\text{minimize}} \left\|A - P_U(A)\right\|_{\text{F}}^2 \tag{2.72}$$

To summarize, the singular value decomposition offers an intuition into the description of coherent structures. Namely, given a data set expressed as vectors that reside in the vector space $\mathbb{R}^n$, when arranged into columns and concatenated into a matrix, the singular value decomposition finds the subspace U such that the projection of the vectors on to the subspace is maximized. In the next section, properties of coherent structures (POD modes) and their subtleties are discussed.

2.3.3 Properties of Coherent Structures

First note that, by definition, coherent structures are assumed to be admissible. This condition is enforced in the definition of the coherent structures. Moreover, by construction, coherent structures are constrained to be orthonormal, this can be seen in the problem definition in equations 2.44 and 2.46. The first two properties of coherent structures are restated from Holmes et al. [61],

Property 1. *Coherent structures satisfy linear boundary conditions.*

Property 2. *Coherent structures are orthonormal to each other, i.e.*

$$\int_{\Omega} \phi_i^{\top}(X)\,\phi_j(X)\,dX = \delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases} \tag{2.73}$$

In section 5.2.1, explains that the POD modes are linear combinations of the snapshots, hence

$$\phi_i = \sum_{j=1}^{N} \alpha_j \mathrm{col}_j(u^{\mathrm{d}}) \tag{2.74}$$

This implies the next property borrowed from Berkooz [2],

Property 3. *If all the snapshots satisfy a linear property, then the POD modes also satisfy the same linear property.*

For example, in incompressible flow the snapshots satisfy the continuity equation and are divergence free. Since divergence is a linear operator, this means that the coherent struc-

tures are also divergence free by the following

$$\begin{aligned} \text{div}\,(\phi_i) &= \text{div}\left(\sum_{j=1}^{N} \alpha_j \text{col}_j(u^{\text{d}})\right) \\ &= \sum_{j=1}^{N} \alpha_j \,\text{div}\left(\text{col}_j(u^{\text{d}})\right) \\ &= 0. \end{aligned} \tag{2.75}$$

Property 4. *If the snapshots are divergence free, then the coherent structures preserve the divergence free property as well,*

$$\textit{div}\,(\phi_i) = 0, \quad i = 1, \ldots, m. \tag{2.76}$$

Other properties of coherent structures require some mathematical manipulation. Equation 2.39 shows that coherent structures are used as a basis to express the fluctuations in fluid flow from the mean. The temporal coordinate is computed as follows

$$a_i(t) = \langle v(X,t)\,,\phi_i(X)\rangle \tag{2.77}$$

Now consider the following

$$\begin{aligned} \langle a_i(t)\,,a_j(t)\rangle &= \int_0^T \Big\langle v(X,t)\,,\phi_i(X)\Big\rangle_X \Big\langle v(X',t)\,,\phi_j(X')\Big\rangle_{X'} \,\text{d}t, \\ &= \left\langle \int_\Omega \int_0^T v(X,t)v^\top(X',t)\,\text{d}t\,\phi_i(X)\,\text{d}X\,,\phi_j(X)\right\rangle_X, \\ &= \left\langle \int_\Omega R(X,X')\,\phi_i(X)\,\text{d}X\,,\phi_j(X')\right\rangle_{X'}, \\ &= \left\langle \lambda_i\phi_i(X')\,,\phi_j(X')\right\rangle_{X'}, \\ &= \lambda_i\delta_{ij}. \end{aligned} \tag{2.78}$$

This shows the next property of coherent structures.

Property 5. *Coherent structures are uncorrelated on average.*

$$\mathbb{E}\left[a_i a_j^\top\right] = \delta_{ij}\lambda_i. \tag{2.79}$$

Another property of coherent structures is realized when equation 2.79 and equation 2.73 are introduced into the two point spatial correlation function in equation 2.41. This reduces the spatial correlation function into a more manageable form.

Property 6. *Given two points $X_1, X_2 \in \Omega$, the two point spatial correlation function simplifies to*

$$R(X_1, X_2) = \sum_{i=1}^{n} \lambda_i \phi_i(X_1)\phi_i^\top(X_2). \tag{2.80}$$

In the previous sections, model-order reduction via projection onto a set of basis functions was described. Moreover, the proper orthogonal decomposition was also described as a method to extract basis function (POD modes) from a data set. In the next section, projection-based model-order reduction onto the basis functions extracted via the POD method is described.

2.4 POD Based Reduced-Order Models

The POD based reduced-order modeling techniques typically follow two steps. In the first step an appropriate set of basis functions are computed from simulation data using the POD method. In the second step, a reduced-order model is created using the basis functions from step one. In step two, the Galerkin projection technique provides the reduced-order model by projecting the Navier-Stokes equations onto the POD basis functions.

The combination of both the POD and Galerkin projection methods produces an elegant solution to generating reduced-order models that have found great success in describing flow mechanics. The details are given in the next section.

2.4.1 POD-Galerkin Projection Reduced-Order Models

After the modes or the coherent structures have been extracted, the Navier-Stokes equations are projected onto a smaller dimensional subspace. The projection is carried out using Galerkin methods which yields a set of ODEs. Consider the Navier-Stokes equation 2.6

$$\frac{\partial U}{\partial t} = \nu \nabla^2 U - (U \cdot \nabla) U - \nabla p, \tag{2.81}$$

and the fluctuation of the flow field variables, which can be decomposed into a sum of orthonormal spatial and temporal modes as shown in equation 2.42 and repeated here

$$\begin{aligned} U(X,t) &= \mathbb{E}\left[U(X,t)\right] + v(X,t), \\ &= \mathbb{E}\left[U(X,t)\right] + \sum_{i=1}^{\infty} \phi_i(X) a_i(t). \end{aligned} \tag{2.82}$$

Without loss of generality, it is assumed that $\mathbb{E}[U(X,t)] = 0$ is a steady solution of equation 2.81, otherwise there will be extra terms.

Inserting equation 2.82 into equation 2.81 results in the following representations of the Navier-Stokes equation

$$\frac{\partial v}{\partial t} = \nu \nabla^2 v - (v \cdot \nabla) v - \nabla p. \tag{2.83}$$

which is a slight variation of equation 2.18. Expanding the fluctuation terms in equation 2.83 yields

$$\begin{aligned} \frac{\partial}{\partial t}\left(\sum_{i=1}^{\infty} \phi_i(X,t) a_i(t)\right) &= \nu \nabla^2 \left(\sum_{i=1}^{\infty} \phi_i(X,t) a_i(t)\right) \\ &\quad - \left(\sum_{i=1}^{\infty} \phi_i(X,t) a_i(t) \cdot \nabla\right)\left(\sum_{i=1}^{\infty} \phi_i(X,t) a_i(t)\right) - \nabla p. \end{aligned} \tag{2.84}$$

Since the spatial modes are assumed to be orthonormal, the new representation of the

Navier-Stokes equation (equation 2.83) is projected onto the basis functions as was previously shown in equation 2.31. The inner product is used to perform the projections onto the $k-$th mode ϕ_k as follows

$$\left\langle \phi_k\left(X\right), \frac{\partial v}{\partial t} \right\rangle_X = \left\langle \phi_k\left(X\right), \nu\nabla^2 v - \left(v\cdot\nabla\right)v - \nabla p \right\rangle_X . \tag{2.85}$$

Due to the nature of the inner product it can be distributed as follows

$$\left\langle \phi_k\left(X\right), \frac{\partial v}{\partial t} \right\rangle_X = \nu\left\langle \phi_k\left(X\right), \nabla^2 v \right\rangle_X - \left\langle \phi_k\left(X\right), \left(v\cdot\nabla\right)v \right\rangle_X - \left\langle \phi_k\left(X\right), \nabla p \right\rangle_X , \tag{2.86}$$

which enables the simplification of each part separately. Starting with the left hand side of equation 2.86, and recalling that for orthonormal modes $< \phi_i , \phi_j >= \delta_{ij}$,

$$\begin{aligned}
\left\langle \phi_k\left(X\right), \frac{\partial}{\partial t} v(X,t) \right\rangle_X &= \left\langle \phi_k\left(X\right), \frac{\partial}{\partial t} \sum_{i=1}^{\infty} \phi_i(X) a_i(t) \right\rangle_X , \\
&= \left\langle \phi_k\left(X\right), \sum_{i=1}^{\infty} \phi_i(X) \frac{\mathrm{d}}{\mathrm{d}t} a_i(t) \right\rangle_X , \\
&= \sum_{i=1}^{\infty} \left\langle \phi_k\left(X\right), \phi_i(X) \frac{\mathrm{d}}{\mathrm{d}t} a_i(t) \right\rangle_X , \\
&= \sum_{i=1}^{\infty} \left\langle \phi_k\left(X\right), \phi_i(X) \right\rangle_X \frac{\mathrm{d}}{\mathrm{d}t} a_i(t), \\
&= \sum_{i=1}^{\infty} \delta_{ki} \frac{\mathrm{d}}{\mathrm{d}t} a_i(t), \\
&= \frac{\mathrm{d}}{\mathrm{d}t} a_k(t).
\end{aligned} \tag{2.87}$$

For the first term in the right hand side of equation 2.86,

$$\begin{aligned}
\left\langle \phi_k(X)\,, \nabla^2 v \right\rangle_X &= \left\langle \phi_k(X)\,, \nabla^2 \sum_{i=1}^{\infty} \phi_i(X) a_i(t) \right\rangle_X, \\
&= \left\langle \phi_k(X)\,, \sum_{i=1}^{\infty} \nabla^2 \phi_i(X) a_i(t) \right\rangle_X, \\
&= \sum_{i=1}^{\infty} \left\langle \phi_k(X)\,, \nabla^2 \phi_i(X) \right\rangle_X a_i(t), \\
&= \sum_{i=1}^{\infty} B_i^k a_i(t).
\end{aligned} \tag{2.88}$$

Moving on to the second term in equation 2.86

$$\begin{aligned}
\left\langle \phi_k(X)\,, (v\cdot\nabla)\, v \right\rangle_X, &= \left\langle \phi_k(X)\,, \left\langle \sum_{i=1}^{\infty} \phi_i(X) a_i(t)\,, \nabla \right\rangle_X \sum_{j=1}^{\infty} \phi_j(X) a_j(t) \right\rangle_X, \\
&= \left\langle \phi_k(X)\,, \sum_{i=1}^{\infty} a_i(t) \left\langle \phi_i(X)\,, \nabla \right\rangle_X \sum_{j=1}^{\infty} \phi_j(X) a_j(t) \right\rangle_X, \\
&= \left\langle \phi_k(X)\,, \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \left\langle \phi_i(X)\,, \nabla \right\rangle_X \phi_j(X) a_i(t) a_j(t) \right\rangle_X, \\
&= \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \left\langle \phi_k(X)\,, \left\langle \phi_i(X)\,, \nabla \right\rangle_X \phi_j(X) \right\rangle_X a_i(t) a_j(t), \\
&= \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} C_{ij}^k a_i(t) a_j(t).
\end{aligned} \tag{2.89}$$

The expression can be further simplified for incompressible flows. Using Stokes' theorem

and the fact that $\text{div}(\phi_k) = 0$, the pressure term is simplified as follows

$$\begin{aligned}
\left\langle \phi_k(X)\,, \nabla p \right\rangle_X &= \iiint_\Omega \phi_k \cdot \nabla p \,\mathrm{d}V, \\
&= \iiint_\Omega \nabla \cdot (p\phi_k) - p\,(\nabla \cdot \phi_k)\,\mathrm{d}V, \\
&= \iiint_\Omega \text{div}\,(p\phi_k) - p\,(\text{div}\,(\phi_k))\,\mathrm{d}V, \\
&= \iint_{\partial\Omega} (p\phi_k \cdot \vec{n})\,\mathrm{d}S, \\
&= 0,
\end{aligned} \tag{2.90}$$

where $\vec{n}$ denotes the vector normal to the boundary surface $\partial\Omega$ pointing outwards. In many cases, the boundary integral vanishes, for example, if the velocity is zero along the boundary, or the case where the domain is finite with no inflow or outflow, or the case if the fluid evolves in a periodic domain.

The projection of the incompressible Navier-Stokes equation onto an infinite set of orthonormal modes as shown in equation 2.86 is now expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t} a_k(t) = \nu \sum_{i=1}^{\infty} B_i^k a_i(t) - \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} C_{ij}^k a_i(t) a_j(t), \quad k = 1, \ldots, \infty, \tag{2.91}$$

which is an infinite set of differential equation. Note, that at this point, no approximation has been done yet, and equation 2.91 is the exact transformation of a PDE into an infinite set of ODEs accomplished by the introduction of an orthonormal basis. When the modes are chosen to be the POD modes, the basis functions become the coherent structures and $a_k(t)$ determines the dynamics of $\phi_k(X)$ by describing its evolution and contribution to the flow.

If the infinite set of differential equations is truncated, the resulting system is a finite approximation of the partial differential equation. This approximation technique is known as the Galerkin approximation first discovered by Walther Ritz. By setting all coefficients of a_k to zero for $k > m$, the infinite set of differential equations is reduced to a finite,

truncated set of differential equations. This change of basis and truncation, is equivalent to a projection onto a finite set of orthonormal modes, as was shown in equation 2.31.

The Galerkin approximation of the Navier-Stokes equation by m orthonormal basis functions yields the projection based reduced-order model shown in equation 2.33 with the following reduced dynamics

$$\frac{\mathrm{d}}{\mathrm{d}t}a_k(t) = \nu \sum_{i=1}^{m} B_i^k a_i(t) - \sum_{i=1}^{m}\sum_{j=1}^{m} C_{ij}^k a_i(t)a_j(t), \quad k = 1, \ldots, m, \tag{2.92}$$

which is a reduced-order model with order m. The reduced-order model is then numerically integrated and the Galerkin approximation of the flow variables is given by

$$v(X,t) \approx v_m(X,t) = \sum_{i=1}^{m} \phi_i(X,t)a_i(t) = \Phi a(t). \tag{2.93}$$

It is important to note that the selection and the order of the orthonormal basis functions plays an important role in the Galerkin approximation. The error of the reduced-order

model is associated with the cost function defined in equation 2.44

$$
\begin{aligned}
\epsilon_m &= \int_0^T \left\| v\left(X,t\right) - v_m(X,t) \right\|_X^2 \, \mathrm{d}t, \\
&= \int_0^T \| \sum_{i=1}^{\infty} \phi_i(X) a_i(t) - \sum_{i=1}^{m} \phi_i(X) a_i(t) \|_X^2 \, \mathrm{d}t, \\
&= \int_0^T \| \sum_{i=m+1}^{\infty} \phi_i(X) a_i(t) \|_X^2 \, \mathrm{d}t, \\
&= \int_0^T \left\langle \sum_{i=m+1}^{\infty} \phi_i(X) a_i(t) \, , \sum_{i=m+1}^{\infty} \phi_i(X) a_i(t) \right\rangle_X \mathrm{d}t, \\
&= \int_0^T \sum_{i=m+1}^{\infty} \sum_{j=m+1}^{\infty} \left\langle \phi_i(X) \, , \phi_j(X) \right\rangle_X a_i^{\star}(t) a_j(t) \, \mathrm{d}t, \\
&= \sum_{i=m+1}^{\infty} \sum_{j=m+1}^{\infty} \int_0^T \delta_{ij} \, a_i^{\star}(t) a_j(t) \, \mathrm{d}t, \\
&= \sum_{i=m+1}^{\infty} \int_0^T a_i^{\star}(t) a_i \, \mathrm{d}t, \\
&= \sum_{i=m+1}^{\infty} \lambda_i.
\end{aligned}
\tag{2.94}
$$

When the POD modes (coherent structures) are selected and ordered according to their corresponding eigenvalues in descending order

$$
\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots , \tag{2.95}
$$

the error ϵ_m of the reduced-order model is minimized. An important property to note is that the error monotonically decreases by adding more POD bases into the reduced-order model. This provides a means of selecting which POD modes to use in creating a reduced-order model. A typical truncation criteria is to include the first m modes such that

$$
\frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \approx 0.9. \tag{2.96}
$$

However, Taira et al. [4] argue that there is no clear indication of what the truncation criteria

should be.

In many circumstances, it can be more helpful to pair the POD with a system identification technique instead of a projection-based model reduction technique. The next section describes how the POD can be used with system identification algorithms.

2.4.2 POD and System Identification Reduced-Order Models

There are several issues that occur from POD-Galerkin reduced-order models. First, for nonlinear systems, there are no guarantees that the reduced-order model is stable, even though the full-order model being approximated is stable [64]. Second, many times the underlying equations that govern the behavior of the full-order model are unknown, therefore it is sometimes desirable to generate a data driven reduced-order model. This masks the need for the knowledge of the underlying physics of the problem at hand.

Typically, a data set is observed over a period of time. The POD is then used to compute the orthonormal basis function, m of which are kept while the rest are truncated. The reduced coordinates can then be computed

$$a_i(t) = \left\langle v(X,t), \phi_i(t) \right\rangle_X, \quad i = 1, \ldots, m. \tag{2.97}$$

This set of reduced coordinates is then coupled with an algorithm of choice to create a data driven reduced-order model.

For example, Issac [37] developed a reduced-order model for the pressure distribution of the AGARD 445.6 wing under structural deformations using the POD as a dimensionality reduction algorithm and the eigen-system realization algorithm [65] (ERA) as a system identification algorithm at different Mach numbers. The method showed good results in predicting the pressure distribution over a deforming structure, hence it can be used for dynamic aeroelastic analysis.

Other system identification techniques such as neural networks, subspace identifica-

tion, Krigging and many other can be used as well. There are several drawbacks with this approach, as the data-driven models can at best only contain the dynamics present in the observations as no knowledge of the physics is incorporated into the model [29].

The scientific community has paired the POD method with a multitude of methods over a wide spectrum of applications. Moreover, in the past few decades, many researchers have adapted the POD method in order to improve its performance. In the next section, a few notable extensions to the POD method are discussed.

2.4.3 Extensions

Initially, Lumley [1] in 1967, introduced the idea of extracting coherent structures in turbulent flows by taking the eigenvalue decomposition of the two point spatial correlation matrix. Sirovich [3] in 1987 found a more numerically tractable way to extract the coherent structures from simulation data. Namely, the proper orthogonal decomposition of a single data matrix, Sirovich named this method the snapshot method. The proper orthogonal decomposition, which can be implemented using the singular value decomposition, is more numerically stable and is much more computationally tractable than the eigenvalue decomposition implementation.

In 1995, Everson and Sirovich [66], developed the Gappy POD algorithm. The algorithm successfully reconstructs full images of a face when 90% of the pixels of an image were missing by applying the POD to a data set of facial images. Even though the reconstructed image is not part of the training data set, the method showed very promising results. In 2004, Bui-Thanh, Damodaran and Willcox [67] applied the Gappy POD in area of aerodynamics in two different ways. First, they showed that they were able to reconstruct aerodynamic data over airfoils when the data set is incomplete. Second, the Gappy POD method was used to perform inverse design, namely, given a desired pressure distribution, the optimal airfoil shape can be determined by interpolation of known designs. LeGresley and Alonso [68] also demonstrated aerodynamic shape optimization using the

POD method.

The use of the POD method for model-order reduction was also exploited for flow control purposes. Constructing controls for systems governed by partial differential equations is a numerically challenging problem. In 1996, Tang et al. [69] were the first to use the POD method to control the flow around a rotating cylinder. In 1999, Kunisch and Volkwein [54] successfully demonstrated the same approach to create a controller for the Burger's equation with one spatial variable.

The POD method was initially demonstrated for creating reduced model pertaining to incompressible flows. In incompressible flow, the pressure terms can be eliminated because pressure only acts to enforce the incompressibility constraint, hence only the velocity flow variables are important. This allows for the use of the traditional inner product without modification at the projection step which gives acceptable results. In 2004, Rowley at al. [70] extended the POD method to compressible flow by using an energy-based generalization of the inner product to perform the Galerkin projection step. This is because in compressible flow, the pressure and enthalpy variables become dynamically important, hence there is a need for an inner product that combines thermodynamic and kinetic variables in a rational way.

In 2009, Chaturantabut and Sorensen [71] developed the discrete empirical interpolation method based on the work of Barrault et al. [72]. Consider the following PDE composed of a linear part $\mathcal{L}(\cdot)$ and a nonlinear part $N(\cdot)$

$$\frac{\partial U}{\partial t}(X,t) = \mathcal{L}\big(U(X,t)\big) + N\big(U(X,t)\big). \tag{2.98}$$

Once a finite difference scheme is applied, the PDE is approximated with the following set of ODEs

$$\frac{\mathrm{d}u^{\mathrm{sd}}}{\mathrm{d}t}(t) = Au^{\mathrm{sd}}(t) + \tilde{N}\left(u^{\mathrm{sd}}(t)\right) \tag{2.99}$$

where $A \in \mathbb{R}^{n\times n}$ and $\tilde{N}(\cdot)$ is a nonlinear function. The matrix A and the nonlinear function

$\tilde{N}(\cdot)$ represent a discrete approximations of $\mathcal{L}(\cdot)$ and $N(\cdot)$, respectively. Using a numerical integration scheme, the ODEs can be simulated to produce snapshots $\{u_i^{\mathrm{d}}\}_{i=1}^{N}$. By applying a Galerkin projection onto V, the following reduced-order model is obtained

$$\frac{\mathrm{d}}{\mathrm{d}t}a(t) = V^\top AVa + V^\top \tilde{N}\left(Va(t)\right). \tag{2.100}$$

The matrix $\tilde{A} = V^\top AV$, is computed once in order to generate the reduced-order model, however the problem resides in computing $V^\top \tilde{N}(Va(t))$. This is because the nonlinearity is a function of $a(t)$, hence it needs to be computed at every iteration. For some reduced-order models, computing the nonlinearity at every iteration can be as computationally expensive as simulating the full order model. Discrete empirical interpolation method (DEIM) attempts to solve this problem by approximating the nonlinear function $\tilde{N}(\cdot)$ by projecting it onto a subspace spanned by a few nonlinear functions given by $w_1, \ldots, w_m$. This results in the following approximation

$$\tilde{N}(Va(t)) \approx Wc\left(a(t)\right). \tag{2.101}$$

where $W = [w_1, \ldots, w_m] \in \mathbb{R}^{n\times m}$, and $c(a(t))$ is the corresponding coefficient vector. It is clear that equation 2.101 has a lot more rows than columns. The reason for that is by choice of m basis functions in matrix W, with the purpose of reducing the computational cost and approximating $\tilde{N}(\cdot)$. This makes equation 2.101 an over-determined system of equations, which needs to be solved in order to determine the correct coefficients $c(a(t))$. This is solved by letting some rows on the left hand side equal to the rows on the right hand side and using a matrix $P = [e_{\rho_1}, \ldots, e_{\rho_m}]$, where e_j is the j-th column of the identity matrix. Hence,

$$P^\top \tilde{N}\left(Va(t)\right) = \left(P^\top W\right) c\left(a(t)\right). \tag{2.102}$$

Which leads to the following approximation

$$\tilde{N}(Va(t)) \approx Wc\left(Va(t)\right) = W\left(P^{\top}W\right)^{-1}P^{\top}\tilde{N}\left(Va(t)\right). \tag{2.103}$$

What remains is the correct selection of the basis functions and matrix P. The matrix W is computed by applying the POD to the matrix of nonlinear snapshots, $[\tilde{N}(u_1^{\mathrm{d}}), \ldots, \tilde{N}(u_N^{\mathrm{d}})]$, and the matrix P is constructed via the DEIM algorithm.

In some cases, symmetries in the flow exist. Sirovich [3] made use of this symmetry to extend and enlarge the data set yielding more accurate modes. It is important that the reduced-order model retain the symmetric behavior of the original system so it reflects the qualitative properties of the original system. This also ensures that the reduced model will inherit other properties such as dissipativity and stability. In 1993, Aubry et al. [63] demonstrated the symmetric POD. This method guarantees that the reduced-order model inherits symmetric properties of the full-order model (PDE) when the solution to the full order model is invariant under finite spatial symmetries. This causes the reduced-order model to preserve dissipative characteristics of the full-order model. Suppose that for a given flow problem, the solution is invariant under a symmetry group G, such that if $U(X,t)$ is a solution, then every $g \circ U(X,t)$ is also a solution to the flow problem $\forall g \in G$. The symmetric POD results in symmetric modes, which are extracted from the symmetric two point correlation function defined as

$$R_s\left(X,X'\right) = \lim_{T\to\infty}\frac{1}{KT}\int_0^T \sum_{g\in G} u\left(\psi_g\left(X\right),t\right) u\left(\psi_g\left(X'\right),t\right)\,\mathrm{d}t, \tag{2.104}$$

where K is the number of elements in G. The symmetric modes are then computed as eigenvectors to the following eigenvalue problem

$$\int_{\Omega} R_s\left(X,X'\right)\phi(X')\,\mathrm{d}X' = \lambda\phi(X). \tag{2.105}$$

There are many more extensions and adaptions to the POD algorithm, however, only a few notable ones are mentioned here. In the next section, some of the limitations of the POD method are discussed.

2.4.4 Limitations

Although the POD has shown promising results in the field of model-order reduction, it does have limitations. One of the limitations of the POD method is the parameter sensitivity of the POD reduced-order models [23]. It is important to note that the POD modes are typically computed using simulations of the flow at specific conditions such as Reynolds number, Mach number, etc., which are called parameters. The POD modes derived are suitable in creating reduced-order models only when the parameters are held constant. If the coherent structures derived under one set of parameters are used to create reduced-order models of fluid systems with another set of parameters, then POD basis functions fail to generate accurate reduced-order models. As shown in equation 2.74, the POD modes are a linear combination of the snapshots. Bui-Thanh et al. [73] note that the span of the POD modes is the span of the snapshots, which determines the quality of the reduced-order model. Research in this area is still active, and more details can be found in the following references [74, 40, 73, 57].

When reduced-order models are created via the POD method, only the first m most energetic modes are retained, while the rest of the modes are disposed. Another shortcoming of the POD method is that sometimes a large number of modes are needed to approximate the full-order model, which does not provide computational savings. Moin and Keefe [41] demonstrated the inefficiency of the POD method in accurately describing fully turbulent models without including a large number of modes.

Rowley et al. [30] note that although the most energetic POD modes do in fact span an optimal subspace for describing a data set, these modes might not be optimal for creating reduced-order models, which is the case for non-normal systems with large transient

growth, such as the ones that often arise in shear flow. This is because low energy features may be critical to the dynamics but do not necessarily best describe the given data set [75]. Taira et al. [4] also note this observation, namely, the POD arranges the modes in order of energy content and not by dynamical importance. For example, in turbulent flows, low energy modes trigger big changes in the flow, hence not including the low energy modes misrepresent the dynamics and the physics of the flow. In these cases, creating a reduced-order model using only the most energetic modes fails to distill and capture the essential dynamics of the flow and as a result misrepresents the dynamics of the original full-order system. Aubry, Lian, and Titi [63] demonstrated that a POD-Galerkin approximation of six POD modes that captured 99.9995% of the total energy did not reproduce the right dynamics of the Kuramoto-Sivashinsky equation. Therefore, an extension of the POD method, the symmetric POD method, which involves the incorporation of the symmetry group is suggested. The method guarantees that the reduced-order model inherits symmetric properties of the full-order model (PDE) when the solution to the full order model is invariant under finite spatial symmetries. This causes the reduced-order model to preserve dissipative characteristics of the full-order model.

Even though the POD method is considered to be the central method for generating reduced-order models for nonlinear systems, it is a linear procedure. It is this linearity that has made the POD method a great success, but also the source of its limitations [2]. The inability of the POD method to capture the essential dynamics may be attributed to the linear nature of the method, i.e. the POD method can only capture dynamics expressed as a linear combination of the snapshots, as revealed by equation 2.74. Tiara et al. [4] also note that the POD method relies on second-order correlations and ignores higher-order correlations.

Some of the limitations of the POD, particularly the inability of the POD method to capture important dynamical features that do not contribute to the system's energy can be alleviated by the balanced truncation method. The balanced truncation, is a technique de-

veloped by Moore that made ground breaking advancements. First presented in his famous papers [76], Moore demonstrated a method for the reducing linear dynamical systems in state space form with strong error guarantees. The method works by finding a similarity transformation that transforms the state space model to one where the more controllable states are also the more observable states. Eliminating states that are less controllable and less observable, results in eliminating states that have less influence on the input-output behavior of the full-order system, resulting in an accurate reduced-order model. Unfortunately, the method of balanced truncation requires the computation of the controllability and observability gramians of the full-order model, which is not computationally tractable for large systems. As the balanced truncation became one of the most popular model reduction methods, different variants of the method arose. A survey of them can be found in the paper by Gugercin and Antoulas [77].

Lall et al. [78, 79] showed how a numerical approximation of the controllability and observability gramians, called the empirical gramians, can be used to find important subspaces of the state space, with respect to the inputs and outputs of the system. Since this method only requires data from simulations of the full-order system to approximate its gramians, the suggested method can be applied to nonlinear systems and hence generalizes the balanced truncation method to nonlinear systems. However, for large systems, the construction of the two empirical gramians is still challenging and, in many cases, numerically impractical [75]. Willcox and Peraire [80] went further, using simulations of the dual system, they are able to obtain low-rank approximations of the gramians without having to compute the matrices themselves. The method readily extends to multiple-input multiple-output systems, however, it becomes intractable when the number of outputs is large. Rowley [75] criticizes this work, showing that it results in inaccurate models and does not actually result in balanced realizations.

Not confusing the previously mentioned method by Willcox and Peraire with the balanced POD, in 2005, Rowley [75] developed the balanced POD (BPOD) method, which

requires adjoint simulations for model reduction. The BPOD method is a computationally tractable method for computing approximate balanced truncations, which produces models that are superior to the POD. This is because the BPOD method captures small-energy perturbations that are highly observable, i.e. contribute to the output. However, the BPOD method is limited to stable, linear systems and relies on the availability of adjoint information, which may not be available, specially with experimental measurements [4]. In fact, most of the previously mentioned methods require the full-order governing equations to generate the reduced-order models by projecting the dynamics onto a reduced subspace. Some of the algorithms even require the adjoint model in order to generate the adjoint information needed.

An algorithm developed by Juang and Papa [65], known as the eigen-system realization algorithm, only requires the impulse response and no full order governing equations to generate the reduced-order model. This is a huge advantage, since for many CFD applications, it is difficult to extract the governing full order equations and it is easier to collect the impulse response from simulation runs. Ma and Rowley [81] also showed that the BPOD method is equivalent to the ERA, however the ERA does not require the system equations and only relies on simulation data. Unfortunately, the ERA results in a reduced-order model that is linear, requires impulse responses, and does not provide modes which can be used to understand the dynamics of the fluid system under investigation.

According to Rowley et al. [4], the dynamic mode decomposition (DMD) decomposes the data into modes, where each mode is associated with a frequency of oscillation and a growth/decay rate. Furthermore, the DMD algorithm has gained popularity due to its customizability and versatility. However, the DMD is unreliable for nonlinear systems and only produces linear reduced-order models with non orthogonal modes. In 2018, Towne, Schmidt, and Colonius [82], established a deep connection between the DMD algorithm the spectral proper orthogonal decomposition [83], showing that the SPOD modes are optimally averaged DMD modes.

Rowley and Dawson [30] provide an excellent review of the balanced truncation, POD, BPOD, ERA and DMD algorithms and a survey of resources demonstrating their application to fluid systems. Moreover, the algorithms mentioned are applied to linearized channel flow and their results are compared. Many other variations and adaptations of the POD have been developed and continue to be developed such as the Split POD, Temporal POD, Joint POD, Spectral POD and others [4].

Attempts in incorporating nonlinear dimensionality reduction (NLDR) algorithms from the field of machine learning (ML) or manifold learning is also gaining popularity in the modal analysis of fluid systems [29]. Such ML algorithms like the local linear embedding (LLE), nonlinear principal component analysis, cluster based dimensionality reduction, neural networks, and others are an active area of research. Brunton, Noack, and Koumoutsakos [29] provide an excellent discussion with a survey of references for machine learning methods applied in fluid analysis. It is interesting to note that the POD can be formulated as a two layer neural network with a linear activation function for its linearly weighted input, that can be trained by stochastic gradient descent [29]. They caution, however, that machine learning algorithms will always provide some answer to any question based on training data, even if the data is not relevant to the question at hand. Furthermore, it is important to consider the interpretability (the degree to which a model may be understood or interpreted by an expert human) of the results and to what degree should machine learning based models be explainable. They also add that it is important to develop and adapt ML algorithms that are physics-informed and physics-consistent. If machine learning algorithms are to be incorporated into the analysis of fluid systems, then machine learning solutions need to be interpretable, explainable, and generalizable.

Lucia, Beran, and Silva [23] mention that the intent of constructing ROM is to provide a quantitative description of the dynamics at a much lower computational cost than that of the original full-order model and to provide a means by which the system dynamics can be readily interpreted. Bakwell and Lumley [84] were one of the first to apply the

POD to analyze turbulent pipe flow. The first POD mode computed revealed a pair of counter-rotating eddies. In 1991, Breuer and Sirovich [85] detail the effectiveness of the POD in extracting the basis functions from simulation data of the two dimensional wave equation. Breuer and Sirovich also demonstrated the robustness of the POD method in its ability to extract the basis functions from simulation data under different resolutions and noise contamination. Berkooz, Holmes and Lumley [2] discuss how the POD method can be used in extracting the essential features of turbulent flow. These early examples demonstrate the effectiveness of the POD method in extracting interpretable flow features that capture the physics of the flow.

At the heart the POD algorithm and its variations is the singular value decomposition applied to data put in a matrix form such as matrix u^{d} in equation 2.54 as show in section 2.3.2. George [5] discusses that the nature of the data does not matter whether it is velocity, pressure or temperature, the integral represented in equation 2.52 will have the appropriate cross-correlation function (equation 2.41). Hence, George argues that it would be interesting to explore the consequences of alternative choices for maximization (equation 2.46).

The importance of modal analysis and reduced-order modeling of fluid systems continues to grow. The POD is important and an active area of research as it serves as the bedrock of fluid modal analysis and reduced-order modeling of fluid systems [4]. The POD method derived by Lumley [1] has deep mathematical roots as an attempt to extract coherent structures that help scientist and engineers understand the nature of fluid flow. The amount of methods, algorithms, and their adaptations is vast and their applications are still not fully explored, which raises the question, is there a need for another method? The purpose of this thesis is to investigate the suggestion by George [5], to explore a fundamentally different alternative to the cross-correlation function, that is better suited for a three-dimensional flow field, without losing the mathematical relevance to coherent structures.

As discussed, the POD is statistically-based, has analytical foundations that supply a

clear understanding of its capabilities and limitations, and provides a rigorous mathematical framework for the extraction of description of coherent structures [2]. It also offers a tool for the construction of low-dimensional dynamical models from the Navier-Stokes equations. The intent of this thesis is to provide a more natural, physics-based framework for treating three-dimensional fluid flow problems while generalizing the proper orthogonal decomposition, hence preserving its favorable features and extending the method to higher dimensional spaces. Hence, the goal of this thesis is to address some of the limitations associated with the POD method while remaining consistent with the Lumley's mathematical representation of coherent structures, consequently furthering the understanding of modal analysis of fluid systems.

The following chapter discusses how replacing the traditional field of real numbers $\mathbb{R}$ with a four dimensional non-commutative division algebra $\mathbb{H}$ known as the quaternion division algebra, provides an avenue to generalize the POD method and address its limitations.

CHAPTER 3
PROBLEM FORMULATION

This thesis builds on the ideas of Lumley [1] and Sirovich [3] in an effort to generalize the POD method, and the notion of coherent structures. In 1967, Lumley provided a mathematical framework for the extraction of persistent patterns, which he termed coherent structures; in 1987, Sirovich provided an implementation of Lumley's method that is computationally tractable. However, little attention has been paid to incorporating quaternions into Lumley's mathematical framework. In recent years, interest in quaternions has been increasing as non-commutative algebras find applications in the areas of science, mathematics, and engineering. However, to the author's knowledge, quaternions have never been investigated in the field of modal analysis and reduced-order modeling of fluid systems. This raises the following research question:

Research Question. *Are there any benefits associated with a approach to the modal analysis and reduced-order modeling of three-dimensional fluid systems?*

The use of quaternions in the description of the flow field variables, generalizes the proper orthogonal decomposition to the quaternion proper orthogonal decomposition, and the concept of coherent structures to quaternion coherent structures. The contribution of this thesis is two fold. Firstly, it seeks to extend the current state of the art by introducing quaternions to the proper orthogonal decomposition in the context of modal analysis and model-order reduction of three-dimensional fluid systems, and investigates the benefits associated with the quaternion approach. Secondly, just like the POD method, the direct implementation of the QPOD method is computationally challenging and intractable for large problems. Hence, a scalable, computationally tractable implementation of the quaternion proper orthogonal decomposition is introduced and tested. The method follows from the work of Sirovich [3], and is termed the quaternion snapshot method.

In order to move forward with this investigation, quaternions and its algebra need to be introduced. To that end, section 3.1 provides an informal introduction to the necessary background information on quaternions. Section 3.2 highlights some of the applications of quaternions in various fields of science and engineering. In section 3.3 the method being investigated in this thesis are developed and presented. Finally, section 3.4 formulates the concepts and the research conducted in this thesis.

3.1 Quaternions

Quaternions, were first discovered by Sir Hamilton [86] in 1843. Just as the complex numbers are a natural extension of the real numbers, so are the quaternion numbers a natural extension of the complex numbers ($\mathbb{R} \subset \mathbb{C} \subset \mathbb{H}$). Quaternions, can also be considered as a special case of octonions, $\mathbb{O} \supset \mathbb{H}$. In 1898, Hurwitz proved that the only normed algebras are the real, complex, quaternion and octonion algebras. Hurwitz theorem is stated here:

Theorem 1. *The only normed division algebras, which are number systems where we can add, subtract, multiply and divide, and which have a norm satisfying*

$$|zw| = |z||w| \tag{3.1}$$

have dimension 1,2,4, or 8.

The aim of this section is to informally introduce quaternions and their important features that will be used to develop the ideas of the methodology. The following introduction contains cherry picked ideas necessary for the remainder of this thesis, and is borrowed from references [87, 88, 89, 90] A more detailed treatment of quaternions can be found in reference [91, 92].

3.1.1 Quaternion Numbers

Definition 1. A quaternion q is an element of the four dimensional normed algebra $\mathbb{H}$ over the real numbers with basis $\{\mathbf{1}, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$. Quaternions have one real part, and three imaginary parts

$$q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k} \tag{3.2}$$

where $a, b, c, d \in \mathbb{R}$ and $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are imaginary units.

The element a is called the real part and $q - a$ is called the vector or imaginary part. A quaternion is said to be pure if it has a null real part $(a = 0)$. Sometimes a quaternion can also be compactly represented as

$$q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k} = (a, q - a). \tag{3.3}$$

Many notations for quaternions exist, more notably is the Cayley-Dickson notation, where a quaternion can be expressed uniquely as

$$q = \alpha + \beta\boldsymbol{j}, \quad \alpha, \beta \in \mathbb{C}, \tag{3.4}$$

where $\alpha = a + b\boldsymbol{i}$ and $\beta = c + d\boldsymbol{i}$. The Cayley-Dickson notation provides an isomorphism between $\mathbb{H}$ and $\mathbb{C}^2$, casting a single quaternion as two complex numbers which will be useful in quaternion matrix processing.

Let $q_1, q_2 \in \mathbb{H}$ be two quaternion numbers given by

$$q_1 = a_1 + b_1\boldsymbol{i} + c_1\boldsymbol{j} + d_1\boldsymbol{k} = (a_1, \vec{u}_1), \tag{3.5}$$

$$q_2 = a_2 + b_2\boldsymbol{i} + c_2\boldsymbol{j} + d_2\boldsymbol{k} = (a_2, \vec{u}_2). \tag{3.6}$$

The addition of two quaternions is given by

$$q_1 + q_2 = (a_1 + a_2) + (b_1 + b_2)\boldsymbol{i} + (c_1 + c_2)\boldsymbol{j} + (d_1 + d_2)\boldsymbol{k}, \tag{3.7}$$

$$= (a_1 + a_2, \vec{u}_1 + \vec{u}_2). \tag{3.8}$$

The multiplication rules for multiplying the quaternion imaginary units are typically summarized as

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -\mathbf{1}. \tag{3.9}$$

The following set of equations explicitly state the remaining multiplication rules

$$\boldsymbol{ij} = -\boldsymbol{ji} = \boldsymbol{k}, \tag{3.10}$$

$$\boldsymbol{ki} = -\boldsymbol{ik} = \boldsymbol{j}, \tag{3.11}$$

$$\boldsymbol{jk} = -\boldsymbol{kj} = \boldsymbol{i}. \tag{3.12}$$

This implies the following quaternion multiplication of q_1 and q_2

$$\begin{aligned} q_1 q_2 =& (a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2) \\ &+ (b_1 a_2 + a_1 b_2 - d_1 c_2 + c_1 d_2)\boldsymbol{i} \\ &+ (c_1 a_2 + d_1 b_2 + a_1 c_2 - b_1 d_2)\boldsymbol{j} \\ &+ (d_1 a_2 - c_1 b_2 + b_1 c_2 + a_1 d_2)\boldsymbol{k}, \\ =& (a_1 a_2 - \vec{u}_1 \cdot \vec{u}_2, a_2 \vec{u}_1 + a_1 \vec{u}_2 + \vec{u}_1 \times \vec{u}_2). \end{aligned} \tag{3.13}$$

If q_1 and q_2 are pure quaternions such that $a_1 = a_2 = 0$, then their product simplifies to

$$q_1 q_2 = (-\vec{u}_1 \cdot \vec{u}_2, \vec{u}_1 \times \vec{u}_2). \tag{3.14}$$

Note that quaternion multiplication is not commutative, i.e.

$$q_1 q_2 \neq q_2 q_1. \tag{3.15}$$

Euler's formula can also be extended to quaternions. This allows for the interpretation of quaternions in terms of modulus and phase. Given a quaternion $q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$, it can also be expressed as

$$q = \rho e^{\zeta\theta} \quad \text{where} \quad \begin{cases} \rho = \sqrt{a^2 + b^2 + c^2 + d^2}, \\ \zeta = \frac{bi+cj+dk}{\sqrt{b^2+c^2+d^2}}, \\ \theta = \arctan\left(\frac{\sqrt{b^2+c^2+d^2}}{a}\right) \end{cases}. \tag{3.16}$$

In this polar formulation, ρ is the modulus of q, ζ is a pure unitary quaternion and θ is the angle between the real part and the three dimensional vector part.

Definition 2. The conjugate of a quaternion, $q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k} = (a, \vec{u})$, is defined as

$$\begin{aligned} \bar{q} &= a - b\boldsymbol{i} - c\boldsymbol{j} - d\boldsymbol{k}, \\ &= (a, -\vec{u}). \end{aligned} \tag{3.17}$$

Definition 3. The norm of a quaternion is given by

$$|q| = \sqrt{q\bar{q}} = \sqrt{\bar{q}q} = \sqrt{a^2 + b^2 + c^2 + d^2}. \tag{3.18}$$

A quaternion q is called a unit quaternion if its norm is 1.

Definition 4. The inverse of a quaternion is defined as

$$q^{-1} = \frac{\bar{q}}{|q|^2}. \tag{3.19}$$

One of the first applications of quaternions was in the modeling of rotation and orientation. This is because a three-dimensional vector $p \in \mathbb{R}^3$ where $p = (p_x, p_y, p_z)$, can be rotated by an angle θ around an axis $e = (e_x, e_y, e_z)$ using quaternion multiplication. In order to accomplish the three-dimensional rotation, define $p^{\mathrm{q}}, q \in \mathbb{H}$ and such that

$$p^{\mathrm{q}} = p_x \boldsymbol{i} + p_y \boldsymbol{j} + p_z \boldsymbol{k}, \tag{3.20}$$

$$q = \cos(\theta/2) + e_x \sin(\theta/2)\boldsymbol{i} + e_y \sin(\theta/2)\boldsymbol{j} + e_z \sin(\theta/2)\boldsymbol{k}. \tag{3.21}$$

The product $p^{\mathrm{q,rot}} = qp^{\mathrm{q}}q^{-1}$ is a pure quaternion and its $\boldsymbol{ijk}$-components represent the rotation of p around the axis e by an angle of θ. The rotated coordinates are given by $p^{\mathrm{rot}} = (p_{\boldsymbol{i}}^{\mathrm{q,rot}}, p_{\boldsymbol{j}}^{\mathrm{q,rot}}, p_{\boldsymbol{k}}^{\mathrm{q,rot}}) \in \mathbb{R}^3$.

A similar procedure exists for rotating four-dimensional vectors using quaternion multiplication. A more detailed discussion on four-dimensional rotations via quaternion multiplication can be found in reference [93].

3.1.2 Quaternion Vectors

Since quaternion multiplication is not commutative, this results in two possible vector spaces over the quaternion scalar division algebra $\mathbb{H}$. For the purpose of this research, only the right vector space over $\mathbb{H}$ needs to be considered. The convention that matrices operate on the left of vectors from the vector space $\mathbb{H}^n$ and scalars on the right was chosen which allows for the recovery of classical matrix calculus rules [89, 90].

Definition 5. A vector space of dimension n, namely $\mathbb{H}^n$, over the division algebra of quaternions $\mathbb{H}$, is a right vector space if $\forall v \in \mathbb{H}^n$ and $\forall \zeta, \mu \in \mathbb{H}$

$$(v\zeta)\mu = v(\zeta\mu). \tag{3.22}$$

When applying the POD method, the snapshots are defined to be elements of a Hilbert space [28]. Hilbert spaces provide the means for projecting vectors onto each other and

hence onto subspaces spanned by vectors; a key enabler for optimization. This motivates the following definition:

Definition 6. A quaternion Hilbert space is a right vector space whose elements are vectors of quaternions, $q = (q_1, \ldots, q_n)^\top \in \mathbb{H}^n$ and $q_i \in \mathbb{H}$. Over the vector space, the inner product between two quaternion vectors $p, q \in \mathbb{H}^n$ is given by

$$\langle p \,,\, q \rangle = p^\star q = \sum_{i=1}^{n} \bar{p}_i q_i, \tag{3.23}$$

where $\star$ denotes the quaternion transposition-conjugate operator.

Definition 7. Two quaternions vectors are said to be orthogonal if $\langle x \,,\, y \rangle = 0$.

The inner product naturally introduces a norm and a metric. The norm of a quaternion vector q is given by

$$\|q\| = \sqrt{\langle q \,,\, q \rangle}, \tag{3.24}$$

and the distance between quaternion vectors p and q is given by

$$d(x, y) = \|x - y\| = \left((x - y)^\star (x - y)\right)^{\frac{1}{2}}. \tag{3.25}$$

Definition 8. The right span $S^{\mathfrak{q}}$ of m quaternion vectors $q_1, \ldots, q_m \in \mathbb{H}^n$ is the set of all their linear combinations

$$\begin{aligned} S^{\mathfrak{q}} &= \text{span}\big(q_1, \ldots, q_m\big) \\ &= \big\{ q \in \mathbb{H}^n \text{ such that } q = \sum_{i=1}^{m} q_i a_i^{\mathfrak{q}},\ \forall a_i^{\mathfrak{q}} \in \mathbb{H} \big\}. \end{aligned} \tag{3.26}$$

The superscript "q" is used to emphasize the fact that $a_i^{\mathfrak{q}}$ are quaternion scalars.

3.1.3 Quaternion Matrices

Definition 9. Given N quaternion vectors, where each quaternion vector $q_i \in \mathbb{H}^n$, a quaternion matrix Q is defined as $Q = [q_1, \ldots, q_N]$.

The quaternion valued matrix defines a new vector space, the vector space of quaternion matrices where $Q \in \mathbb{H}^{n \times N}$. Let $\alpha, \beta \in \mathbb{H}$ be quaternion scalars, $p, q \in \mathbb{H}^N$ be quaternion vectors and $Q \in \mathbb{H}^{n \times N}$ a quaternion matrix, the following linearity of Q is satisfied

$$Q(p\alpha + q\beta) = (Qp)\alpha + (Qq)\beta. \tag{3.27}$$

For quaternion matrices, the $\star$ operation represents the quaternion conjugate-transpose operation. This involves the regular transpose operation as is done for real matrices, followed by the quaternion conjugation, as shown in equation 3.17, applied to all of the matrix entries. Given two quaternion matrices $\Phi \in \mathbb{H}^{n \times m}$ and $V \in \mathbb{H}^{m \times N}$, the following is true

$$(\Phi V)^\star = V^\star \Phi^\star, \tag{3.28}$$

$$(V^\star)^{-1} = (V^{-1})^\star. \tag{3.29}$$

The quaternion conjugate-transpose is used in the following definitions:

Definition 10. A quaternion matrix $Q \in \mathbb{H}^{n \times n}$ is hermitian if $Q = Q^\star$.

Definition 11. A quaternion matrix $Q \in \mathbb{H}^{n \times n}$ is unitary if $QQ^\star = Q^\star Q = \mathrm{I}$.

Definition 12. The Frobenius norm of a quaternion matrix $Q \in \mathbb{H}^{n \times N}$ is given by

$$\|Q\|_{\mathrm{F}} = \sqrt{\mathrm{trace}\left(Q^\star Q\right)}. \tag{3.30}$$

3.1.4 Quaternion Matrix Algebra

Due to the non-commutative nature of quaternions, two formulations for the quaternion eigenvalue eigenvector problem are possible, they are the right eigenvalue problem and the left eigenvalue problem.

Definition 13. The right eigenvalue equation for a quaternion matrix $Q \in \mathbb{H}^{n\times n}$ is

$$Q\phi^{\mathfrak{q}} = \phi^{\mathfrak{q}}\lambda^{\mathfrak{q}}, \tag{3.31}$$

where, $\phi^{\mathfrak{q}} \in \mathbb{H}^n$ is called the eigenvector and $\lambda^{\mathfrak{q}} \in \mathbb{H}$ is called the eigenvalue. The superscript "q" is used to emphasize that the eigenvector and eigenvalue are quaternion, which result from quaternion operations.

Swapping the position of the eigenvector and the eigenvalue in equation 3.31 results in $Q\phi^{\mathfrak{q}} = \lambda^{\mathfrak{q}}\phi^{\mathfrak{q}}$. This is known as the left eigenvalue equation. Due to the non-commutative nature of quaternions, both left and right eigenvalue problems yield different results. However, there are many issues with the left eigenvalue formulation that do not occur with the right eigenvalue formulation [87]. Furthermore, the theory of the right eigenvalue problem is more developed and more widely adopted by others [88], therefore it will be the choice in this thesis as well.

Next, it will be shown how the quaternion eigenvalue decomposition (QEVD) extends the spectral theorem of real matrices to quaternion matrices. The following theorems and their proofs can be found in reference [87, 88].

Theorem 2. *If (λ^q, ϕ^q) is a right eigenpair of $Q \in \mathbb{H}^{n\times n}$, then $(w^{-1}\lambda^q w, \phi^q w)$ is also a right eigenpair of Q, for all nonzero $w \in \mathbb{H}$.*

Proof.

$$Q(\phi^{\mathfrak{q}}w) = (Q\phi^{\mathfrak{q}})w = (\phi^{\mathfrak{q}}\lambda^{\mathfrak{q}})w = (\phi^{\mathfrak{q}}w)(w^{-1}\lambda^{\mathfrak{q}}w). \quad \blacksquare \tag{3.32}$$

This implies that for quaternion matrices, every eigenvalue $\lambda^{\mathfrak{q}}$ belongs to an entire set of eigenvalues called a similarity orbit $\theta(\lambda^{\mathfrak{q}})$, and every eigenvector belongs to an entire set of eigenvectors, which will be defined as the kaleidoscope set $\kappa(\lambda^{\mathfrak{q}})$.

Definition 14. Let $(\lambda^{\mathfrak{q}}, \phi^{\mathfrak{q}})$ be a right eigenpair of a quaternion matrix $Q \in \mathbb{H}^{n\times n}$. The similarity orbit of an eigenvalue $\theta(\lambda^{q})$ is defined as

$$\theta(\lambda^{\mathfrak{q}}) = \left\{w^{-1}\lambda^{\mathfrak{q}}w : w \in \mathbb{H}, w \neq 0\right\}. \tag{3.33}$$

Definition 15. Let $(\lambda^{\mathfrak{q}}, \phi^{\mathfrak{q}})$ be a right eigenpair of a quaternion matrix $Q \in \mathbb{H}^{n\times n}$. The kaleidoscope set of an eigenvalue $\lambda^{\mathfrak{q}}$ is defined as

$$\kappa(\lambda^{\mathfrak{q}}) = \left\{\phi^{\mathfrak{q}}w \in \mathbb{H}^{n} : w \in \mathbb{H}, w \neq 0\right\}. \tag{3.34}$$

It is easy to note that the similarity orbits and kaleidoscope sets do not intersect, i.e. $\theta(\lambda_i^{\mathfrak{q}}) \cap \theta(\lambda_j^{\mathfrak{q}}) = \emptyset$, and $\kappa(\lambda_i^{\mathfrak{q}}) \cap \kappa(\lambda_j^{\mathfrak{q}}) = \emptyset$, when $i \neq j$. Moreover, since multiplying $\lambda^{\mathfrak{q}}$ with a quaternion number w on the right and w^{-1} on its left, $w^{-1}\lambda^{\mathfrak{q}}w$ is a rotation of λ^{q} in a four dimensional space without any scaling. Therefore similarity orbits can be expressed as a conjugacy class of $\lambda^{\mathfrak{q}}$,

$$\theta(\lambda^{\mathfrak{q}}) = \left\{\bar{w}\lambda^{\mathfrak{q}}w : w \in \mathbb{H}, \|w\| = 1\right\}. \tag{3.35}$$

Theorem 3. *If $Q \in \mathbb{H}^{n\times n}$ is hermitian, then every right eigenvalue of Q is real. Moreover, the similarity orbit of each eigenvalue collapses to a single real number $\theta(\lambda_i^{q}) = \{\lambda_i^{q} \in \mathbb{R}\}$.*

The full proof of theorem 3 can be found in reference [88]. However, it is easy to see how the entire similarity orbit of $\lambda^{\mathfrak{q}}$ collapses to a single number if and only if $\lambda^{\mathfrak{q}} \in \mathbb{R}$. This is because all quaternion numbers commute with the real numbers, hence $w^{-1}\lambda^{\mathfrak{q}}w =$

$\lambda^{\mathfrak{q}} w^{-1} w = \lambda^{\mathfrak{q}}$ and $\theta(\lambda^{\mathfrak{q}}) = \{\lambda^{\mathfrak{q}} \in \mathbb{R}\}$. However, even when the similarity orbit collapses to a single real number $\lambda^{\mathfrak{q}}$, the kaleidoscope set does not collapse and the quaternion eigenvectors still belong to an infinite set of eigenvectors $\kappa(\lambda^{\mathfrak{q}})$.

Theorem 4. *If $\alpha, \beta \in \kappa(\lambda^q)$ such that $\|\alpha\| = \|\beta\|$, then there exists $w \in \mathbb{H}$ such that*

$$\alpha = \beta w, \tag{3.36}$$

where $\|w\| = 1$.

Proof. By definition, eigenvectors that belong to the same kaleidoscope set are related to each other via multiplication with a quaternion number $w \in \mathbb{H}$ as follows

$$\alpha = \beta w, \tag{3.37}$$

$$\implies \|\alpha\| = \|\beta w\|, \tag{3.38}$$

$$\implies \|\alpha\| = \|\beta\| \|w\|. \tag{3.39}$$

Since $\|\alpha\|$ equals to $\|\beta\|$, the $\|w\| = 1$. ■

Thus vectors with equal norms in a kaleidoscope set are also related to each other by rotating all their entries in four dimensional space via multiplication with a unit quaternion.

Theorem 3 ensures that all singular values of a quaternion matrix are real numbers. This finally leads to the spectral theorem or the quaternion eigenvalue decomposition (QEVD) for quaternion Hermitian matrices

Theorem 5. *If $Q \in \mathbb{H}^{n \times n}$ is hermitian, then there are matrices $D^q \in \mathbb{R}^{n \times n}$ and $U^q \in \mathbb{H}^{n \times n}$ such that U^q is a unitary matrix, D^q is diagonal, and*

$$Q = U^q D^q U^{q\star}. \tag{3.40}$$

The spectral theorem guarantees the diagonalization of hermitian matrices. Moreover, the matrix U^{q} is unitary which implies that its columns are orthogonal to one another. This paves the way for the quaternion singular value decomposition of quaternion matrices (QSVD).

Theorem 6. *If $Q \in \mathbb{H}^{n\times N}$ is of rank m, then there exists two quaternion unitary matrices $U^q \in \mathbb{H}^{n\times m}$ and $V^q \in \mathbb{H}^{m\times N}$ such that*

$$Q = U^q \Sigma^q V^{q\star}, \tag{3.41}$$

where $\Sigma^q \in \mathbb{R}^{m\times m}$ is a real diagonal matrix with non-negative entries. $U^q \in \mathbb{H}^{n\times n}$ contains the left quaternion singular vectors and $V^q \in \mathbb{H}^{N\times N}$ contains the right quaternion singular vectors of Q. Moreover,

$$\textit{QSVD}(Q) = \textit{QEVD}(QQ^\star). \tag{3.42}$$

The existence of the quaternion singular value decomposition allows for a definition of the rank of quaternion matrices in terms of its singular values.

Definition 16. The rank of a quaternion matrix Q is equal to the number of its nonzero singular values.

The quaternion singular value decomposition can also be expressed in summation form, which motivates the following definition:

Definition 17. Let $Q \in \mathbb{H}^{n\times N}$, the rank-$m$ approximation of Q is defined as

$$Q_m = \sum_{i=1}^{m} u_i^{\mathrm{q}} v_i^{\mathrm{q}\star} \sigma_i^{\mathrm{q}} \tag{3.43}$$

where the singular values are sorted in descending order, $\sigma_1 \geq \sigma_2 \geq \ldots$.

Theorem 7. *If Q_m is the rank-m approximation of $Q \in \mathbb{H}^{n\times N}$, then*

$$\epsilon = \|Q - Q_m\|_F \leq \|Q - M\|_F \tag{3.44}$$

for all quaternion matrices $M \in \mathbb{H}^{n\times N}$ with rank m. Furthermore,

$$\begin{aligned} \epsilon^2 &= \mathit{trace}\big((Q - Q_m)^\star (Q - Q_m)\big), \\ &= \sum_{i=m+1}^{\mathit{min}(n,N)} \big(\sigma_i^q(Q)\big)^2 = \sum_{i=m+1}^{\mathit{min}(n,N)} \lambda_i^q(Q) \end{aligned} \tag{3.45}$$

3.1.5 Quaternion Singular Value Decomposition

In 1997, Zhang [91] proved the existence of the singular value decomposition of a quaternion matrix, however, his work did not show a direct algorithm for computing such a decomposition. In the same paper, Zhang also showed an isomorphism between quaternion matrices and their representation over the complex field known as the complex adjoint matrix. Mehta [94] used the isomorphism to compute the eigenvalue decomposition (QEVD) of quaternion matrices by first casting them into complex matrices. Using the same isomorphism, Le Bihan [95] introduced the first direct algorithm capable of computing the singular value decomposition of quaternion matrix (QSVD). For a given quaternion matrix $Q \in \mathbb{H}^{n\times N}$, the method requires the calculation of the singular value decomposition of a complex matrix $Q^{\mathrm{c}} \in \mathbb{C}^{2n\times 2N}$ with complex entries. Sangwine and Le Bihan [96] developed another QSVD algorithm based on the bidiagonalization of quaternion matrices using the quaternion Householder transformation.

Using the QR factorization of quaternion matrices [97], Le Bihan and Sangwine [98] extended the Jacobi algorithm to directly compute the singular value decomposition of quaternion matrices. Not only is the Jacobi algorithm more computationally efficient, it was also shown to produce more accurate results [98, 99]. An implementation of the quaternion

singular value decomposition in MATLAB was developed by Sangwine [100] and made available to the public.

These observations refine the research objectives of this thesis as follows:

Research Objective. *Explore the use of quaternions in the context of modal analysis and reduced-order modeling of three-dimensional fluid systems.*

In the next section, a brief overview on the applications of quaternions in science and engineering is presented. In addition, a derivation of the quaternion description of the three-dimensional incompressible Navier-Stokes equations is outlined.

3.2 Quaternions in Science and Engineering

The POD method has been successful in many areas of science and engineering. In fact, in 1991, Sirovich and Kirby [101, 102] applied the POD method to gray scale images of human faces for the purpose of characterizing human faces. In 2003, Le Bihan and Sangwine [103, 104], extended the gray scale image decomposition methods to include color images. By expressing the RGB components of pixels as a quaternion numbers, a $N \times M$ color image is represented as a pure quaternion image

$$s(x, y) = r(x, y)\boldsymbol{i} + g(x, y)\boldsymbol{j} + b(x, y)\boldsymbol{k}, \tag{3.46}$$

where $r(x, y), g(x, y)$, and $b(x, y)$ are the red, green, and blue components respectively for the pixel at position (x, y). The application of quaternion POD to color images has shown great results [90] and continues to be an active area of research.

In order to compress color images using the POD algorithm, the color components at every pixel are separated and then stacked on top of each other, just like a fluid flow problem, as explained in section 2.3.2. Xu et al. [90] mention that this procedure is unsatisfactory because the correlations among the RGB channels are not properly captured. However, by using quaternions, each color pixel can be treated in a holistic manner. Thus,

the relationship between the color components is better captured by the quaternion singular value decomposition algorithm and less information is lost.

The concept of using a quaternion for signal processing is first introduced in 2004 by Le Bihan and Mars [89] where the quaternion singular value decomposition is applied to quaternion signals. In a vibration test, the measured time series for X,Y, and Z components of acceleration were encoded as a quaternion signal $s(m)$ at different locations,

$$s(m) = a_x(m)\boldsymbol{i} + a_y(m)\boldsymbol{j} + a_z(m)\boldsymbol{k}, \quad m = 1, \ldots, M. \tag{3.47}$$

Then QSVD was used to analyze the signal and the result was compared to the traditional POD method. The authors report the QSVD to be more efficient.

The success of quaternions is not limited to the compression of color images and time signals. Indeed Kou and Xia [105] have surveyed several fields that have benefited from applying quaternions. For example, in computer graphics, quaternions have great advantages over real numbers in modeling rotation and orientation. This is due to the limitation of Euler angles, known as the gimbal lock phenomenon, which completely disappears when quaternions are used to compute the rotations. For this reason many robotics application have adopted the quaternion framework. Kou and Xia [105] also mention how quaternions have found applications in mathematics, kinematic modeling, fluid mechanics, and quantum mechanics.

3.2.1 Quaternion Representation of the Navier-Stokes Equation for Incompressible Flow

In 2002, Gibbon [106] introduced a quaternion formulation of the three-dimensional Euler equations. In 2005, Postnikov and Stepanova [107] built on Gibbon's work to show the incompressible Navier-Stokes equation 2.5, formulated with only the quaternion variables and quaternion operators. Postnikov and Stepanova [107] used the equation of motion of vorticity as the starting point for their derivation of a quaternion representation of the

incompressible Navier-Stokes equation. The equation of motion for vorticity, $\omega = \nabla \times U$, is given by

$$\frac{\partial \omega}{\partial t} = \nabla \times (\omega \times U) + \frac{1}{R}\nabla^2 \omega. \tag{3.48}$$

The velocity quaternion is defined as follows

$$\begin{aligned} U^{\mathrm{q}}(x, y, z, t) &= u(x, y, z, t)\boldsymbol{i} + v(x, y, z, t)\boldsymbol{j} + w(x, y, z, t)\boldsymbol{k} \\ &= \big(0, U(x, y, z, t)\big) \end{aligned} \tag{3.49}$$

and the quaternion derivative as $D = (0, \nabla)$. The following definition of the vorticity quaternion, Ω, is given by

$$DU^{\mathrm{q}} = (0, \nabla)(0, U) = (-\nabla \cdot U\ ,\ \nabla \times U) = (0\ ,\ \nabla \times U) = \Omega. \tag{3.50}$$

Define $F = \frac{1}{2}\left(\Omega U^{\mathrm{q}} - U^{\mathrm{q}}\Omega\right)$, and define the Laplace operator in quaternion form as $D^2 = (0, \nabla)(0, \nabla) = (-\nabla^2, 0)$, the equation of motion for vorticity becomes

$$\frac{\partial \Omega}{\partial t} = \langle DF \rangle - \frac{1}{2} D^2 \Omega \tag{3.51}$$

where $\langle DF \rangle = (DF - \overline{DF})/2$.

The quaternion formulation of the three-dimensional Navier-Stokes shows that using quaternions to represent the velocity field of a three-dimensional fluid system is not an arbitrary one, but a natural one. In fact, using a quaternion representation of the Navier-Stokes equation is in its own right a dimensionality reduction. This is because the number of equations and variables are reduced by two thirds.

The next section describes the quaternion proper orthogonal decomposition and two of its implementations. The first implementation directly computes the quaternion singular value decomposition and is called the quaternion direct implementation. However, the second implementation is indirect, and is introduced for the first time in the work of this

thesis. The idea behind the second implementation is follows from the work of Sirovich [3] and hence it is named the quaternion snapshot implementation.

3.3 Quaternion Proper Orthogonal Decomposition for Three-Dimensional Fluid System

This section, demonstrates how the quaternion proper orthogonal decomposition is applied to three-dimensional velocity data collected from experimental measurements or numerical simulations of fluid systems. The approach introduces quaternions into the mathematical framework suggested by Lumley [1] in his attempt to extract coherent structures. Incorporating quaternions offers a natural extension to Lumley's POD method that is better suited to describe three-dimensional problems. The quaternion approach preserves the mathematical framework/process suggested by Lumley for the extraction of coherent structures, which makes this approach a generalization of Lumley's work and not just another machine learning algorithm that is blindly applied to a large data set for the sake of feature extraction.

In this section, two numerical implementations of the quaternion proper orthogonal decomposition are outlined, they are:

1. Quaternion direct implementation
2. Quaternion snapshot implementation

The quaternion direct implementation is a direct computation of the QPOD on a data set, which just like the traditional POD method, is computationally intractable for large data sets. The quaternion snapshot implementation is an indirect implementation of the QPOD inspired by Sirovich [3]. Both implementations of the QPOD yield the same results, but the quaternion snapshot method is computationally tractable while the direct quaternion method is not. This allows for the scalability of the QPOD method to large data sets as would be encountered in modern day CFD problems.

In section 2.3.2, the standard procedure for applying the POD method is described in details. The process outlines how the collected flow field velocity data are arranged into three matrices

$$U^{\mathrm{d}} = \begin{bmatrix} u_1^1 & u_1^2 & \dots & u_1^N \\ \vdots & \vdots & \dots & \vdots \\ u_n^1 & u_n^2 & \dots & u_n^N \end{bmatrix}, V^{\mathrm{d}} = \begin{bmatrix} v_1^1 & v_1^2 & \dots & v_1^N \\ \vdots & \vdots & \dots & \vdots \\ v_n^1 & v_n^2 & \dots & v_n^N \end{bmatrix}, W^{\mathrm{d}} = \begin{bmatrix} w_1^1 & w_1^2 & \dots & w_1^N \\ \vdots & \vdots & \dots & \vdots \\ w_n^1 & w_n^2 & \dots & w_n^N \end{bmatrix}, \tag{3.52}$$

where $U^d, V^d, W^d \in \mathbb{R}^{n\times N}$, n represents the number of grid points, N represents the number of snapshots (time instances where data is collected), the superscript represents the snapshot instance, and the subscript represents the grid number. The matrices are then concatenated to create the snapshot matrix u^{d}, as shown in equation 2.54.

However, defining the velocity quaternion, as shown in equation 3.49, which is repeated here

$$U^{\mathrm{q}}(x,y,z,t) = u(x,y,z,t)\boldsymbol{i} + v(x,y,z,t)\boldsymbol{j} + w(x,y,z,t)\boldsymbol{k}, \tag{3.53}$$

allows for the quaternion representation of the data collected. This is done by constructing the quaternion snapshot matrix $u^{\mathrm{dq}} \in \mathbb{H}^{n\times N}$ as follows

$$u^{\mathrm{dq}} = \begin{bmatrix} (0,u_1^1,v_1^1,w_1^1) & (0,u_1^2,v_1^2,w_1^2) & \cdots & (0,u_1^N,v_1^N,w_1^N) \\ (0,u_2^1,v_2^1,w_2^1) & (0,u_2^2,v_2^2,w_2^2) & \cdots & (0,u_2^N,v_2^N,w_2^N) \\ \vdots & \vdots & \dots & \vdots \\ (0,u_n^1,v_n^1,w_n^1) & (0,u_n^2,v_n^2,w_n^2) & \cdots & (0,u_n^N,v_n^N,w_n^N) \end{bmatrix}, \tag{3.54}$$

where $(i,j)^{\text{th}}$ entry is a single quaternion number, which represents the spatially and temporally discretized flow field variables given by node i at snapshot j. The matrix u^{dq} can also be expressed using the data matrices defined in the set of equations 3.52 as follows

$$u^{\mathrm{dq}} = U^{\mathrm{d}}\boldsymbol{i} + V^{\mathrm{d}}\boldsymbol{j} + W^{\mathrm{d}}\boldsymbol{k}. \tag{3.55}$$

The norm of the quaternion matrix u^{dq} is given by

$$\|u^{\mathrm{dq}}\|_{\mathrm{F}}^2 = \|U^{\mathrm{d}}\|_{\mathrm{F}}^2 + \|V^{\mathrm{d}}\|_{\mathrm{F}}^2 + \|W^{\mathrm{d}}\|_{\mathrm{F}}^2. \tag{3.56}$$

Lumley defined coherent structures as solutions to the minimization problem stated in equation 2.44. In this work, the quaternion coherent structures $\{\phi_i^{\mathrm{q}}\}_{i=1}^m$ are defined to be the solutions to the following quaternion minimization problem

$$\begin{aligned} \underset{\{\phi_i^{\mathrm{dq}}\}_1^m}{\text{minimize}} \quad & \left\| u^{\mathrm{dq}} - \sum_{i=1}^{m} \phi_i^{\mathrm{q}}(X) a_i^{\mathrm{q}}(t) \right\|_{\mathrm{F}} \\ \text{subject to} \quad & \left\|\phi_i^{\mathrm{q}}(X)\right\|_{\mathrm{F}} = 1, \quad \forall i = 1, \ldots, m. \end{aligned} \tag{3.57}$$

As stated in theorem 7, the solution to the quaternion minimization problem is given by the rank-m approximation of u^{dq}, which is computed via the quaternion singular value decomposition of u^{dq}, or the following eigenvalue decomposition

$$u^{\mathrm{dq}} u^{\mathrm{dq}\star} \phi_i^{\mathrm{q}}(X) = \phi_i^{\mathrm{q}}(X) \lambda_i^{\mathrm{q}}. \tag{3.58}$$

Theorem 5 guarantees $\phi_i^{\mathrm{q}}(X)$ are orthogonal to one another.

Currently, the quaternion proper orthogonal decomposition is implemented via the quaternion direct implementation. In the work of this thesis, the quaternion snapshot implementation is developed in order to provide a computationally friendly approach of applying the QPOD method to large data sets. In the next sections, the details of both implementations are presented.

3.3.1 The Quaternion Direct Method

Given an appropriate data set $U^{\mathrm{d}}, V^{\mathrm{d}}, W^{\mathrm{d}} \in \mathbb{R}^{n \times N}$, the quaternion snapshot matrix u^{dq} is created as shown in equation 3.54 or equation 3.55. The direct implementation performs the QPOD by taking the eigenvalue decomposition of the quaternion two point spatial cor-

relation matrix defined as follows

$$R^{\mathrm{dq}} = \frac{1}{N} u^{\mathrm{dq}} u^{\mathrm{dq}\star}, \tag{3.59}$$

where $R^{\mathrm{dq}} \in \mathbb{H}^{n\times n}$. The matrix R^{dq} is the quaternion version of $R^{\mathrm{d}} \in \mathbb{R}^{(n\times 3)\times(n\times 3)}$ which was used to compute the POD method as outlined in section 2.3.2. Expanding the terms, the matrix $u^{\mathrm{dq}} u^{\mathrm{dq}\star}$ is given by

$$\begin{aligned} u^{\mathrm{dq}} u^{\mathrm{dq}\star} &= \left(U^{\mathrm{d}}\boldsymbol{i} + V^{\mathrm{d}}\boldsymbol{j} + W^{\mathrm{d}}\boldsymbol{k}\right)\left(U^{\mathrm{d}}\boldsymbol{i} + V^{\mathrm{d}}\boldsymbol{j} + W^{\mathrm{d}}\boldsymbol{k}\right)^{\star} \\ &= \left(U^{\mathrm{d}}U^{\mathrm{d}\top} + V^{\mathrm{d}}V^{\mathrm{d}\top} + W^{\mathrm{d}}W^{\mathrm{d}\top}\right) \\ &\quad + \left(W^{\mathrm{d}}V^{\mathrm{d}\top} - V^{\mathrm{d}}W^{\mathrm{d}\top}\right)\boldsymbol{i} \\ &\quad + \left(U^{\mathrm{d}}W^{\mathrm{d}\top} - W^{\mathrm{d}}U^{\mathrm{d}\top}\right)\boldsymbol{j} \\ &\quad + \left(V^{\mathrm{d}}U^{\mathrm{d}\top} - U^{\mathrm{d}}V^{\mathrm{d}\top}\right)\boldsymbol{k}. \end{aligned} \tag{3.60}$$

Equation 3.60 was also verified using another approach shown in the appendix A.1.

Hence, the discretized eigenvalue problem becomes

$$R^{\mathrm{dq}}\phi_i^{\mathrm{q}} = \phi_i^{\mathrm{q}}\lambda_i^{\mathrm{q}}, \tag{3.61}$$

and the eigenvalues and eigenvectors of the matrix R^{dq} are computed. The components associated with every quaternion eigenvector can be extracted as follows

$$\phi_i^{\mathrm{rq}} = \mathrm{real}\left(\phi_i^{\mathrm{q}}\right), \tag{3.62}$$

$$\phi_i^{\mathrm{iq}} = \mathrm{imag}_{\boldsymbol{i}}\left(\phi_i^{\mathrm{q}}\right), \tag{3.63}$$

$$\phi_i^{\mathrm{jq}} = \mathrm{imag}_{\boldsymbol{j}}\left(\phi_i^{\mathrm{q}}\right), \tag{3.64}$$

$$\phi_i^{\mathrm{kq}} = \mathrm{imag}_{\boldsymbol{k}}\left(\phi_i^{\mathrm{q}}\right). \tag{3.65}$$

The eigenvectors ϕ_i^{q} are known as the quaternion coherent structures or the QPOD modes.

Since the size of matrix $R^{\mathrm{dq}} \in \mathbb{H}^{n\times n}$ can be tremendously large, the eigenvalue decomposition of such a matrix can be an impossible task due to computational resource limitations.

Note that the quaternion eigenvalue decomposition of $u^{\mathrm{dq}}u^{\mathrm{dq}\star}$ can also be stated as a quaternion singular value decomposition of u^{dq} as follows

$$\mathrm{QSVD}(u^{\mathrm{q}}) = \mathrm{QEVD}(u^{\mathrm{q}}u^{\mathrm{q}\star}). \tag{3.66}$$

This implementation of the QPOD method is named the quaternion direct implementation. Note that the quaternion singular values of u^{dq}, will be denoted as $\{\sigma_i^{\mathrm{q}}(u^{\mathrm{dq}})\}_i$, and that $(\sigma_i^{\mathrm{q}}(u^{\mathrm{dq}}))^2 = \lambda_i(u^{\mathrm{dq}}u^{\mathrm{dq}\star}), \forall i = 1\dots N$. Note that $u^{\mathrm{dq}}u^{\mathrm{dq}\star}$ is hermitian, and theorem 3 states that the eigenvalues of a hermitian quaternion matrix are real which implies $\sigma_i^{\mathrm{q}}, \lambda_i^{\mathrm{q}} \in \mathbb{R}$.

3.3.2 The Quaternion Snapshot Method

The quaternion direct method, which implements the QPOD, suffers from the same limitations as the direct implementation of the POD. As CFD problems become more complex, CFD grids become finer resulting in a large number of nodes. This makes the direct method intractable as the matrix $R^{\mathrm{dq}} = u^{\mathrm{dq}}u^{\mathrm{dq}\star} \in \mathbb{R}^{n\times n}$ is very large and performing a singular value or eigenvalue decomposition on such a matrix is not practical due computational resource limitations. In this section, the quaternion snapshot implementation is outlined. Th quaternion snapshot implementation addresses the computational limitations associated with the quaternion direct implementation, much like the snapshot implementation introduced by Sirovich [3] addresses the issues associated with the direct POD implementation.

Given the quaternion snapshot matrix u^{dq} as defined in equation 3.54, consider the quaternion right eigenvalue problem of $u^{\mathrm{dq}\star}u^{\mathrm{dq}}$

$$u^{\mathrm{dq}\star}u^{\mathrm{dq}}\psi_i^{\mathrm{q}}(X) = \psi_i^{\mathrm{q}}(X)\gamma_i^{\mathrm{q}}. \tag{3.67}$$

The superscript "q" is used to emphasize that the eigenvalues and eigenvectors are quaternion values and quaternion vectors. It is important to always keep that in mind since quaternions are non-commutative. However, $u^{\mathrm{dq}^\star}u^{\mathrm{dq}}$ is hermitian, i.e. $(u^{\mathrm{dq}^\star}u^{\mathrm{dq}})^\star = u^{\mathrm{dq}^\star}u^{\mathrm{dq}}$ and theorem 3 states that the eigenvalues of a hermitian quaternion matrix are real which implies $\gamma_i^{\mathrm{q}} \in \mathbb{R}$. Multiplying u^{dq} on both sides of equation 3.67 from the left results in

$$u^{\mathrm{dq}}u^{\mathrm{dq}^\star}u^{\mathrm{dq}}\psi_i^{\mathrm{q}}(X) = u^{\mathrm{dq}}\psi_i^{\mathrm{q}}(X)\gamma_i^{\mathrm{q}}. \tag{3.68}$$

Regrouping and inspecting the terms leads to the desired result

$$\left(u^{\mathrm{dq}}u^{\mathrm{dq}^\star}\right)\underbrace{\left(u^{\mathrm{dq}}\psi_i^{\mathrm{q}}(X)\right)}_{\phi_i^{\mathrm{q}}(X)} = \underbrace{\left(u^{\mathrm{dq}}\psi_i^{\mathrm{q}}(X)\right)}_{\phi_i^{\mathrm{q}}(X)}\gamma_i^{\mathrm{q}}. \tag{3.69}$$

where $\gamma_i^q = \lambda_i^q$.

Therefore the QPOD modes can be recovered as follows

$$\phi_i = u^{\mathrm{dq}}\psi_i\frac{1}{\sqrt{(\gamma_i)}}. \tag{3.70}$$

In matrix form, this is written as

$$\Phi = u^{\mathrm{dq}}\Psi\Gamma^{-\frac{1}{2}}, \tag{3.71}$$

where $\Psi = [\psi_1, \ldots, \psi_n]$, and Γ is a diagonal matrix with γ_i along the diagonal.

The next section, outlines the work of this thesis, which investigated the use of the quaternion proper orthogonal decomposition in the context of modal analysis and reduced-order modeling of three-dimensional fluid systems. In addition, an investigation on the quaternion snapshot implementation is also outlined.

3.4 Research Contributions

The purpose of this research is to investigate the use of quaternions for the modal analysis and reduced-order modeling of three dimensional fluid systems. The following list of observations motivate the endeavor:

Observation 1. *Just like complex numbers has contributed to many areas of science, evidence show that quaternion numbers have also contributed to many scientific fields [105], particularly in problems associated with three or four dimensions. This suggests that there are potential benefits in the modal analysis and the reduced-order modeling of fluid systems where quaternions have not been considered yet.*

Observation 2. *In section 3.2.1, the three dimensional incompressible Navier-Stokes equations are stated purely in terms of quaternion variables and quaternion functions only [107]. The quaternion approach treats the velocity variables (u, v, w) as a single holistic variable which is in itself a reduction of variables. This suggests that a quaternion representation of the fluid velocity variables is not an ad-hoc one.*

Observation 3. *Many existing reduced-order modeling methods are capable of approximating fluid systems. However, there is a need for methods that can better distill the physics of fluid systems by capturing the essential dynamics present in the full-order model. Such methods will help scientists and aircraft engineers further their understanding and intuition of fluid flow [23].*

Observation 1 motivates the investigation of using quaternions in the context of modal analysis and reduced-order modeling of fluid systems. Observations 2 and 3 suggest that a quaternion formulation of model-order reduction for fluid systems would be fruitful, and would provide insights in understanding fluid flow; however there is no evidence to support such a claim. This gap in research leads to the following research question:

Research Question. *Are there any benefits associated with a quaternion approach to the*

modal analysis and reduced-order modeling of three-dimensional fluid systems?

A complete investigation of this question would require a large effort, because there are different ways to incorporate quaternions in fluid flow [106, 107] and many more reduced-order modeling methods. As a first step towards the stated research question, this thesis will focus on incorporating quaternions to a benchmark method. The following observations from the literature helps narrow down the research problem.

Observation 4. *The POD is the benchmark method for the modal analysis and reduced-order modeling of fluid systems [2, 30, 4].*

Observation 5. *The cross-correlation function in the POD method does not account for the dimensionality of the spatial domain or the source of the data. George [5] suggested the exploration of the consequences to alternative choices for the cross-correlation function.*

Observation 6. *The singular value decomposition applies to quaternion matrices in the same way it applies to complex and real matrices, i.e. the quaternion singular value decomposition generates optimal rank-m approximations of quaternion matrices [91, 89].*

Observations 4 identifies the POD method as the benchmark method, while observation 5 suggests a starting point for the investigation. Moreover, observation 6 suggests that such an endeavor is possible as the mathematical framework needed has been developed. Therefore, the scope of this research is narrowed down and the research question is restated as:

Research Question 1. *How would a quaternion approach to the POD method in the context of modal analysis and reduced-order modeling of three-dimensional fluid systems compare to the traditional POD method?*

As previously mentioned, little attention has been devoted to the use of quaternions in the modal analysis of fluid systems. However, the following observations provide some evidence that will help in formulating an informed guess to research question 1.

Observation 7. *The quaternion proper orthogonal decomposition has shown promising*

results in color image compression [108, 90, 109, 103, 104] and three-dimensional signal processing of acceleration measurements [89]. These problems share a strong parallel with reduced-order modeling of fluid systems because of the following common attributes:

1. *Discretization of a spatial domain. In the case of images, a spatial discretization is realized as a grid of pixels associated with an image. In the case of the acceleration measurements, sensors are placed in specific locations in space. For fluid systems, a spatial domain is discretized by generating a CFD grid or by placing sensors that measure velocity at locations of interest.*

2. *Three components of information are associated with every discretized point. In the case of color images, the components of information are the intensities of the red, green, and blue colors associated with every pixel. In the case of the signal processing of acceleration measurements, the components of information are the XYZ components of acceleration (a_x, a_y, and a_z) at every location of measurement. For fluid systems, the three components of information are the three components of velocity (u, v, and w) associated with each grid node or location of measurement.*

Observation 8. *The QPOD has shown promising results in color image processing, even though the color components of an image are not bound to each other under any relationship. In the case for fluid systems, the velocity components relate to each other under an explicit set of mathematical rules, the Navier-Stokes equations, which can be stated purely in terms of quaternions.*

Observations 7 and 8 suggest that a quaternion approach to the POD method (QPOD) for three-dimensional fluid systems would be superior to the traditional POD. This leads to the following hypothesis to research question 1:

Hypothesis 1. *If a quaternion approach is used in the context of model-order reduction of three dimensional fluid systems, then the QPOD method will out perform the traditional POD method.*

Due to the lack of research in the literature, there is no evidence to support the main hypothesis. Therefore a number of experiments need to be devised that would put this hypothesis to the test.

In the work of this thesis, the quaternion velocity flow field variable is defined as shown in equation 3.49, which is repeated here

$$U^{\mathrm{q}}(x, y, z, t) = u(x, y, z, t)\boldsymbol{i} + v(x, y, z, t)\boldsymbol{j} + w(x, y, z, t)\boldsymbol{k}. \tag{3.72}$$

Using quaternions to express the velocity flow field variables results in a fundamental change to the POD method. However, the quaternion formulation preserves the mathematical treatment and rigor used by Lumley [1] to derive the concept of coherent structures. The consequences of equation 3.72 are summarized in table 3.1, which highlights the similarities and differences between the POD and the QPOD.

POD	**QPOD**
$u(x,y,z,t), v(x,y,z,t), w(x,y,z,t)$	$u(x,y,z,t)\boldsymbol{i} + v(x,y,z,t)\boldsymbol{j} + w(x,y,z,t)\boldsymbol{k}$
$\underset{\{\phi_i\}_1^m}{\text{minimize}} \quad \left\|u^{\mathrm{d}} - \sum_{i=1}^m \phi_i(X)a_i(t)\right\|_{\mathrm{F}}$ subject to $\quad \|\phi_i(X)\| = 1,\ \forall i = 1,\ldots,m.$	$\underset{\{\phi_i^{\mathrm{q}}\}_1^m}{\text{minimize}} \quad \left\|u^{\mathrm{dq}} - \sum_{i=1}^m \phi_i^{\mathrm{q}}(X)a_i^{\mathrm{q}}(t)\right\|_{\mathrm{F}}$ subject to $\quad \|\phi_i^{\mathrm{q}}(X)\| = 1,\ \forall i = 1,\ldots,m.$
$u^{\mathrm{d}} = \begin{bmatrix} {U^{\mathrm{d}}}^\top & {V^{\mathrm{d}}}^\top & {W^{\mathrm{d}}}^\top \end{bmatrix}^\top \in \mathbb{R}^{(n\times 3)\times N}$	$u^{\mathrm{dq}} = U^{\mathrm{d}}\boldsymbol{i} + V^{\mathrm{d}}\boldsymbol{j} + W^{\mathrm{d}}\boldsymbol{k} \in \mathbb{H}^{n\times N}$
$u^{\mathrm{d}}{u^{\mathrm{d}}}^\top \phi_i(X) = \phi_i(X)\lambda_i$	$u^{\mathrm{dq}}{u^{\mathrm{dq}}}^\star \phi_i^{\mathrm{q}}(X) = \phi_i^{\mathrm{q}}(X)\lambda_i^{\mathrm{q}}$
$\{\lambda_i = (\sigma_i)^2\}_{i=1}^m$ $\phi_i^u = \mathrm{row}_{j=1}^{n}(\phi_i)$ $\phi_i^v = \mathrm{row}_{j=n+1}^{n\times 2}(\phi_i)$ $\phi_i^w = \mathrm{row}_{j=n\times 2+1}^{n\times 3}(\phi_i)$	$\{\lambda_i^{\mathrm{q}} = (\sigma_i^{\mathrm{q}})^2\}_{i=1}^m$ $\phi_i^{\mathrm{rq}} = \mathrm{real}\,(\phi_i^{\mathrm{q}})$ $\phi_i^{\mathrm{iq}} = \mathrm{imag}_{\boldsymbol{i}}\,(\phi_i^{\mathrm{q}})$ $\phi_i^{\mathrm{jq}} = \mathrm{imag}_{\boldsymbol{j}}\,(\phi_i^{\mathrm{q}})$ $\phi_i^{\mathrm{kq}} = \mathrm{imag}_{\boldsymbol{k}}\,(\phi_i^{\mathrm{q}})$
$u_m^{\mathrm{d}} = \sum_{i=1}^m \phi_i(X)a_i(t)$ $a_i(t) = <\phi_i(X), U(X,t)>$	$u_m^{\mathrm{dq}} = \sum_{i=1}^m \phi_i^{\mathrm{q}}(X)a_i^{\mathrm{q}}(t)$ $a_i^{\mathrm{q}}(t) = <\phi_i^{\mathrm{q}}(X), U^{\mathrm{q}}(X,t)>$

Table 3.1: A summary highlighting the similarities and differences between the POD method and the QPOD method. Introducing quaternions at the root of the problem description trickles down to differences in formulation, computation, solution, and approximation.

As discussed in section 2.4, the general strategy for reduced-order modeling of fluid systems involves two steps. In the first step, basis functions are extracted from the flow data, this is also known as feature extraction, dimensionality reduction, or pattern recognition. When the POD algorithm is used to perform step one, the modal analysis community named the resulting basis functions as coherent structures (POD modes). In the second step, a reduced-order modeling technique (such as Galerkin projection, ERA, neural nets, etc.) is used to generate the reduced-order model.

This thesis focuses on comparing the POD and the QPOD methods which are applied at step one. The thesis does not focus on reduced-order modeling techniques typically applied at step two. In other words, this thesis compares the coherent structures extracted via the

POD against the quaternion coherent structures extracted via the QPOD. Since the coherent structures extract information pertaining to the flow's dynamics, coherent structures that are richer in information and flow dynamics will consequently generate better reduced-order models. Therefore, a first step towards a quaternion approach is undertaken by showing that the quaternion coherent structures are superior to coherent structures. This directly contributes to the modal analysis and reduced-order modeling of fluid systems.

Observation 9. *The effectiveness of modal analysis and reduced-order modeling of fluid systems lies in the extraction of basis functions with rich information pertaining to the problem at hand. Creating basis functions that better capture the dynamics and the physics of a fluid system, contributes to a more accurate modal analysis and to a superior reduced-order modeling capability of three-dimensional fluid systems.*

In order to find evidence that would test the main hypothesis, four experiments were devised. In the first two experiments, the POD and the QPOD methods are applied to hypothetical data sets. In the third and fourth experiments, the POD and the QPOD methods are applied to two different fluid systems. However, before the experiments are described in more details, the next section discusses the metrics that will be used to assess the performance of each method.

3.4.1 Supporting Arguments

In this section, a number of important questions that will be used to answer research question 1 are addressed. In order to assess if the QPOD method is better than the POD method, it is important to develop a metric that can measure the performance of each method.

Research Question 1.1. *How is the performance of each method quantified?*

As stated before, the POD method computes the optimal functions $\{\phi_i(X)\}_{i=1}^{m}$ that solves

the following minimization problem

$$\begin{aligned} &\underset{\{\phi_i\}_{i=1}^m}{\text{minimize}}\ J_m = \left\| u^{\mathrm{d}} - \sum_{i=1}^{m} \phi_i(X) a_i(t) \right\|_{\mathrm{F}} \\ &\text{subject to } \left\|\phi_i(X)\right\|_{\mathrm{F}} = 1,\ \forall i = 1, \ldots, m. \end{aligned} \tag{3.73}$$

Hence, given a set of coherent structures $\{\phi_i\}_{i=1}^m$, the following performance metric is suggested

$$\begin{aligned} \epsilon_m^{\mathrm{r}} &= \frac{J_m}{\|u^{\mathrm{d}}\|_{\mathrm{F}}} \times 100, \\ &= \frac{\|u^{\mathrm{d}} - \sum_{i=1}^m \phi_i(X) a_i(t)\|_{\mathrm{F}}}{\|u^{\mathrm{d}}\|_{\mathrm{F}}} \times 100, \\ &= \frac{\|u^{\mathrm{d}} - u_m^{\mathrm{d}}\|_{\mathrm{F}}}{\|u^{\mathrm{d}}\|_{\mathrm{F}}} \times 100. \end{aligned} \tag{3.74}$$

On the other hand, the QPOD method attempt to find a set of basis functions such that $\{\phi_i^{\mathrm{q}}(X)\}_{i=1}^m$ solves the following minimization problem

$$\begin{aligned} &\underset{\{\phi_i^{\mathrm{q}}\}_{i=1}^m}{\text{minimize}}\ J_m^{\mathrm{q}} = \left\| u^{\mathrm{dq}} - \sum_{i=1}^{m} \phi_i^{\mathrm{q}}(X) a_i^{\mathrm{q}}(t) \right\|_{\mathrm{F}} \\ &\text{subject to } \left\|\phi_i^{\mathrm{q}}(X)\right\|_{\mathrm{F}} = 1,\ \forall i = 1, \ldots, m. \end{aligned} \tag{3.75}$$

Hence, given a set of quaternion coherent structures $\{\phi_i^{\mathrm{q}}(X)\}_{i=1}^m$, their performance is measured by

$$\begin{aligned} \epsilon_m^{\mathrm{q}} &= \frac{J_m^{\mathrm{q}}}{\|u^{\mathrm{dq}}\|_{\mathrm{F}}}, \\ &= \frac{\|u^{\mathrm{dq}} - \sum_{i=1}^m \phi_i^{\mathrm{q}}(X) a_i^{\mathrm{q}}(t)\|_{\mathrm{F}}}{\|u^{\mathrm{dq}}\|_{\mathrm{F}}} \times 100, \\ &= \frac{\|u^{\mathrm{dq}} - u_m^{\mathrm{dq}}\|_{\mathrm{F}}}{\|u^{\mathrm{dq}}\|_{\mathrm{F}}} \times 100. \end{aligned} \tag{3.76}$$

This leads to the following hypothesis:

Hypothesis 1.1. *The normalized errors, ϵ_m^r and ϵ_m^q, capture the performance of each*

method in approximating the original data set u^d and u^{dq} respectively.

Research Question 1.2. *How is the performance of each method compared to the other?*

It is important to note that the matrix u^{d} and u^{dq} contain the same data just represented differently as highlighted in table 3.1. Moreover, equation 2.55 and equation 3.56 imply $\|u^{\mathrm{d}}\|_{\mathrm{F}} = \|u^{\mathrm{dq}}\|_{\mathrm{F}} = L$, which suggests the following metric as a performance measure that compares the two methods

$$\epsilon_m = \epsilon_m^{\mathrm{r}} - \epsilon_m^{\mathrm{q}} \tag{3.77}$$

Hypothesis 1.2. *The metric ϵ_m compares the performance of the two methods.*

When $\epsilon_m > 0$, then the QPOD method out performs POD method. When $\epsilon_m < 0$, then the POD method out performs QPOD method.

Now that metrics that capture the performance of the POD and the QPOD methods are established, it would be helpful to quantify how different are the POD modes compared to the QPOD modes. One reason for finding a function that measures the extent to how different the POD and the QPOD solutions are, is to identify situations where both methods have the same performance (i.e. $\epsilon_m = 0$), but the solutions (i.e. POD modes and QPOD modes) are different. This motivates the following research question:

Research Question 1.3. *How is the relationship between the POD results and the QPOD results quantified?*

The solution of the POD method is given as a set of singular values $\{\sigma_i\}_{i=1}^n$, and a set of POD modes $\{\phi_i(X)\}_{i=1}^n$ (coherent structures). Similarly, the solution of the QPOD method is given as a set quaternion singular values $\{\sigma_i^{\mathrm{q}}\}_{i=1}^n$ (which turn out to be real numbers as stated in theorem 3), and a set of QPOD modes $\{\phi_i^{\mathrm{q}}(X)\}_{i=1}^n$ (quaternion coherent structures). The following quantity $\|\Phi - \Phi^{\mathrm{q}}\|$ seems like a reasonable suggestion that captures the size of the difference between the POD modes and the QPOD modes. However, such an expression is ill-posed with numerous issues, most notable is that the dimensions of Φ

and Φ^{q} do not agree.

In 2019, Denton et al. [110] proved that the eigenvectors of a matrix can always be computed using only the eigenvalues of the same matrix. In fact, Denton et al. provide an explicit formula for computing the eigenvectors of a matrix from its eigenvalues. This implies that the POD modes are in fact a function of the singular values $\{\sigma_i^2 = \lambda_i\}_{i=1}^n$.

Observation 10. *The POD modes $\{\phi_i(X)\}_{i=1}^n$ are a function of the singular values $\{\sigma_i^2 = \lambda_i\}_{i=1}^n$ and the explicit mapping is given by*

$$|\phi_{i,j}(X)|^2 = \frac{\prod_{k=1}^{n-1}\left(\lambda_i(u^d {u^d}^\top) - \lambda_k(M_j)\right)}{\prod_{k=1;k\neq i}\left(\lambda_i(u^d {u^d}^\top) - \lambda_k(u^d {u^d}^\top)\right)} \tag{3.78}$$

where $\phi_{i,j}$ is the j^{th} entry of the i^{th} POD mode, and M_j is the $(n-1 \times n-1)$ minor formed by deleting the j^{th} row and column of matrix $u^d {u^d}^\top$.

Hence, any change in the singular values $\{\sigma_i\}_{i=1}^n$ will reflect as a change in the POD modes. This means that all the information associated with the POD solution is completely contained with the singular values. In other words, no two solutions can have the same singular values but different POD modes. There is very strong evidence showing that the same relationship (equation 3.78) holds for quaternion hermitian matrices as well. However, a proof is not available in the literature.

Observation 10 indicates that any difference in the POD results and QPOD results can be measured using only the singular values. This suggests the following distance metric $\eta(\cdot) \in \mathbb{R}$ given as the sum of the squared difference between the ordered sets of singular values as follows

$$\eta = \sum_{i=1}^{n} (\sigma_i^{\mathrm{q}} - \sigma_i)^2 \tag{3.79}$$

Algebraic manipulations of equation 3.79 reveals the following

$$\begin{aligned}
\eta &= \sum_{i=1}^{n} \left(\sigma_i^q - \sigma_i\right)^2, \\
&= \sum_{i=1}^{n} \Bigg(\left(\sigma_i^q\right)^2 + \left(\sigma_i\right)^2 - 2\sigma_i^q\sigma_i \Bigg), \\
&= \sum_{i=1}^{n} \left(\sigma_i^q\right)^2 + \sum_{i=1}^{n} \left(\sigma_i\right)^2 - 2\sum_{i=1}^{n} \sigma_i^q\sigma_i, \\
&= \|u^{\mathrm{dq}}\|_{\mathrm{F}}^2 + \|u^{\mathrm{d}}\|_{\mathrm{F}}^2 - 2\sum_i \sigma_i^q\sigma_i, \\
&= 2L^2 - 2\sum_i \sigma_i^{\mathrm{q}}\sigma_i.
\end{aligned} \tag{3.80}$$

The distance metric is normalized with respect to the norm of the data matrices such that

$$\begin{aligned}
\eta_{\mathrm{n}} &= \frac{\eta}{2L^2} \\
&= 1 - \frac{1}{L^2}\sum_i \sigma_i^{\mathrm{q}}\sigma_i.
\end{aligned} \tag{3.81}$$

Normalizing the distance metric makes it more intuitive as $0 \leq \eta_{\mathrm{n}} \leq 1$, and results in a distance metric that is independent of the amplitude of the flow field in a given data set, which allows for the comparison of η_n to data sets with different scales. This leads to the following hypothesis:

Hypothesis 1.3. *The distance function η_n measures the difference between the information captured in $\{\phi_i(X)\}_{i=1}^n$ and the information captured $\{\phi_i^q(X)\}_{i=1}^n$.*

When $\eta_n = 0$, then the POD modes and the QPOD modes are the same and contain the same information. However, larger values of η_{n} indicate a larger deviation between the POD modes and the QPOD modes.

3.4.2 Secondary Contributions

In section 2.3.2, it was noted that the snapshot method introduced by Sirovich [3] was a key enabler for the scalability of the POD method. This is because the direct implementation of the POD requires the eigenvalue decomposition of the covariance matrix $u^{\mathrm{d}}{u^{\mathrm{d}}}^{\top} \in \mathbb{R}^{(n\times 3)\times(n\times 3)}$, a very large matrix in modern day CFD problems. The same issue persists with the quaternion direct implementation of the QPOD method. This is because the quaternion direct implementation of the POD requires the eigenvalue decomposition of $u^{\mathrm{dq}}u^{\mathrm{dq}^{\star}} \in \mathbb{H}^{n\times n}$. This leads to the following research question:

Research Question 2. *Is the quaternion proper orthogonal decomposition scalable?*

In section 3.3.2, the same algebraic trick introduced by Sirovich [3] was used to derive the quaternion snapshot implementation. The snapshot method introduced by Sirovich was a key enabler in the application of the POD method to large data sets such as the ones encountered in modern day fluid problems.

Observation 11. *The snapshot implementation developed by Sirovich contributed to the scale-ability of the POD method.*

This suggests the following hypothesis:

Hypothesis 2. *If the quaternion snapshot implementation is used, then the QPOD method is scalable.*

There are two more observations worth investigating. The quaternion singular value decomposition of a quaternion matrix u^{dq} results in the following

$$\mathrm{QSVD}(u^{\mathrm{dq}}) = \Phi^{\mathrm{q}}\Sigma^{\mathrm{q}}A^{\mathrm{q}}. \tag{3.82}$$

where the matrix Φ is a quaternion matrix. Even though $\mathrm{real}(u^{\mathrm{dq}}) = 0$, it is not necessary that $\mathrm{real}(\Phi^{\mathrm{q}}) = 0$. This leads to the following:

Observation 12. *Every quaternion mode $\phi_i^q(X)$ is made up of four components instead of three, which are its real part $\phi_i^{rq}(X)$, and its three imaginary parts $\phi_i^{iq}(X)$, $\phi_i^{jq}(X)$, and $\phi_i^{kq}(X)$, as shown in section 3.1.*

This is unexpected as there are only three components associated with every POD mode, where every component decomposes the flow in a certain direction. This observation challenges the established understanding of coherent structures.

Moreover, it was shown in section 3.1.4 in theorem 2, that every eigenvector $\phi_i^{\mathrm{q}}(X)$ of a hermitian matrix belongs to an infinite set of eigenvectors, $\kappa(\lambda_i^{\mathrm{q}})$. Since a multiplication with a quaternion is a rotation in four dimensional space, the quaternion modes can be rotated.

Observation 13. *Every QPOD mode $\phi_i^q(X)$, belongs to an infinite set of eigenvectors, all of which have the same eigenvalue λ_i^q and relate to each other via a four-dimensional rotation. The set is named the kaleidoscope set and is given as*

$$\kappa\big(\lambda_i^q\big) = \{\phi_i^q w : w \in \mathbb{H} \textit{ and } (\lambda_i^q, \phi_i^q) \textit{ is a right eigenpair}\}. \tag{3.83}$$

Observations 12 and 13 are a consequence of using a quaternion approach to the POD method. These observations raise interesting questions as they are not observed in the POD case. In the next section, the four experiments are outlined that test the validity of hypothesis 1 and 2.

3.4.3 Experiments

With the distance metric η and the performance measure ϵ_m defined, it is possible to perform experiments that answer research question 1. However, providing a full answer to research question 1 would require comparing the POD and the QPOD on thousands of three-dimensional fluid data sets. Typically, data pertaining to fluid flow is very large.

Hence, comparing the performances of the POD method and the QPOD method over thousands of such data sets would require vast amounts of computational resources, access to databases where the data is stored in a systematic fashion, and a lot of time. In order to circumvent this issue, four experiments are designed.

In the first experiment a thousand hypothetical data sets are generated. Then both the POD and the QPOD methods are applied to every randomly generated data set. The results are then analyzed, compared, and contrasted. This experiment was done to screen for relationships, correlations, and to discover unexpected results that may stand out. Moreover, it provides the means to describe and compare the average performance of each method.

Experiment 1. *Generate a thousand data sets and apply both the POD and the QPOD methods to each data set. Compare the distance metric η and the performance metric ϵ_m.*

In the second experiment, two hypothetical data sets, data set A and data set B, are generated using the optimization routine `fmincon` available in `MATLAB`. Data sets A and B are generated as follows:

1. Data Set A : find matrices $U^{\mathrm{d}}_{\mathrm{max}}, V^{\mathrm{d}}_{\mathrm{max}}$, and $W^{\mathrm{d}}_{\mathrm{max}}$ that maximize the distance η.

2. Data Set B : find matrices $U^{\mathrm{d}}_{\mathrm{min}}, V^{\mathrm{d}}_{\mathrm{min}}$, and $W^{\mathrm{d}}_{\mathrm{min}}$ that minimizes the distance η.

The POD and the QPOD were then applied to the data sets, and the results were examined.

Experiment 2. *Apply the POD and the QPOD to two special data sets, data set A and data set B. Data set A maximize η while data set B minimizes η.*

In the third experiment, the POD and the QPOD were finally applied to a data set pertaining to a three-dimensional fluid system. The data was obtained experimentally by Jenna Eppink [111, 112], a researcher at the NASA Langley research facility. The data set collected consisted of three-dimensional measurements of the flow over a forward-facing step. The measurement were taken for the purpose of understanding how excrescences (surface imperfections) affect laminar flow and cause the flow to transition. The POD and

the QPOD were then applied and the results were compared.

Experiment 3. *Apply the POD and the QPOD on measurements obtained for a three-dimensional flow over a forward facing step.*

Finally, the fourth experiment was applied to a data set obtained via numerical simulation (CFD) of the flow around a three-dimensional cylinder at different Reynolds numbers. The data set is large, and the quaternion snapshot implementation was used to test for the following:

1. Test the scalability of the quaternion proper orthogonal decomposition.

2. Test the performance of the POD and the QPOD methods.

3. Look for any parameter (Reynolds number) dependence in the performance of QPOD method.

Experiment 4. *Apply the POD and the QPOD via the quaternion snapshot implementation to three-dimensional data obtained via numerical simulation of the flow around a three-dimensional cylinder at different Reynolds numbers.*

Figure 3.1 summarizes the research contributions of this thesis. Chapter 4 will describe in details how experiments one, two, three, and four were carried out.

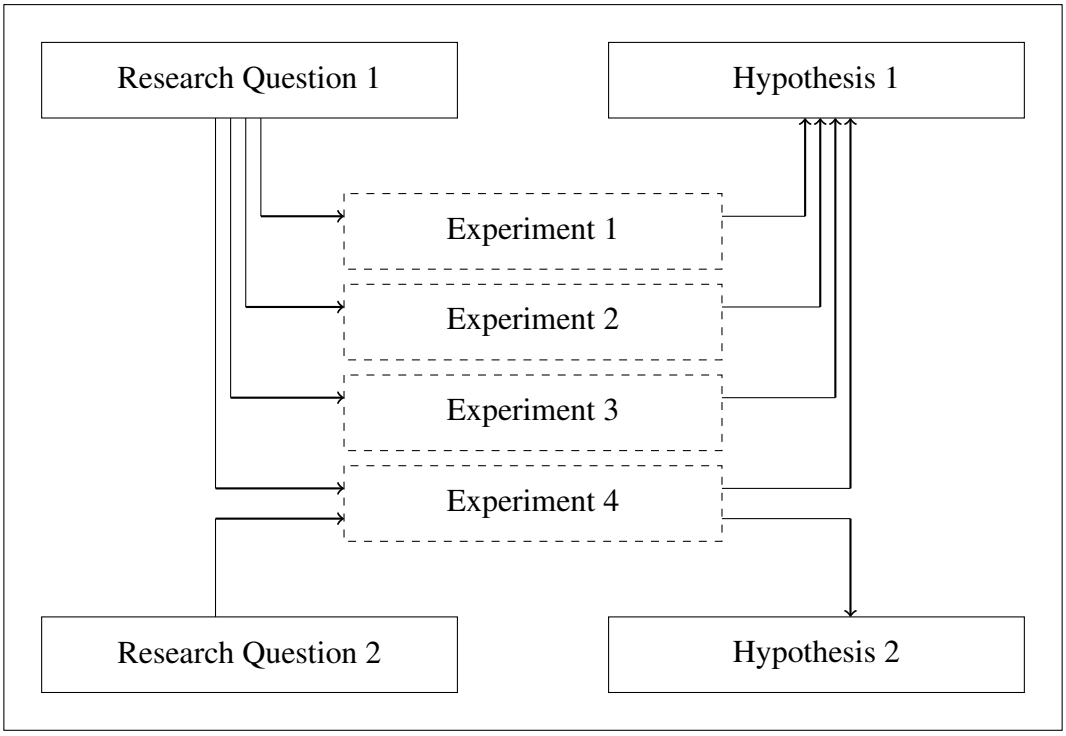


Figure 3.1: Summary of the research questions, hypotheses, experiments.

CHAPTER 4

METHODOLOGY

4.1 Experiments

The current investigation involved four experiments that compare the POD method to the QPOD method. The following is a summary of the data sets used for each experiment:

- **Experiment One:** A thousand data sets were generated via a random number generator using `MATLAB`. The data resembles a hypothetical three-dimensional flow field with five grid nodes observed over the span of five time steps.
- **Experiment Two:** Two special data sets were generated using `MATLAB`'s optimization routine such that η_n is maximized for one and minimized for the other.
- **Experiment Three:** Stereo particle image velocimetry (SPIV) measurements for the flow over a forward facing step in a two foot by three foot channel.
- **Experiment Four:** Flow around a three dimensional cylinder was simulated at different Reynolds numbers and the data was carefully processed, stored, and collected.

By applying the POD and the QPOD methods on a large data set, the first experiment addresses the average performance of each method. However, in the second experiment, a correlation identified in experiment one is confirmed. The third and fourth experiments are used to compare the POD and the QPOD methods as they apply to fluid systems. However, the fourth experiment serves two extra purpose. First, it tests the scalability of the QPOD method to large data sets via the quaternion snapshot implementation (hypothesis 1). Second, it compares the performance of the POD and QPOD methods on a data set obtained for a three-dimensional fluid system. Moreover, experiment four also provides a means for a preliminary analysis that explores any parametric relationship between the Reynolds

number and the performance of each method. The next four sections address the details pertaining to each experiment.

4.2 Experiment One

Comparing the POD and the QPOD methods to a single data set does not provide sufficient evidence to indicate an advantage for a quaternion approach to model order reduction of three-dimensional fluid systems. Hence, the idea behind experiment one is to apply both methods to a large number of data sets in order to:

1. Observe the average performance of each algorithm.

2. Discover relationships, correlations, or trends.

Applying the POD and the QPOD to such a data set would require large amounts of computational resources; moreover, such a data set is not readily available.

To that end, a hypothetical data set that has the attributes of a three dimensional fluid system is created via `MATLAB`. The size of each data set was limited since the purpose of this experiment is not to test the scalability of the methods. By construction, the data set resembles a three-dimensional flow field with five grid nodes observed over the span of five time steps. It is recognized that the size of the data being five by five seems like an arbitrary choice, however, the effect of the data size on the performance of the methods is not under investigation. As a preliminary attempt, the size was chosen to be small because large data sets can be time consuming. Future research should further investigate the effect of the data size on the performance of the methods, but that is not within the scope of this research.

4.2.1 Experiment Details

Algorithm 1 outlines the function used to create the three data matrices $(U^{\mathrm{d}}, V^{\mathrm{d}}, W^{\mathrm{d}}) \in \mathbb{R}^{n \times N}$ using `MATLAB`'s random number generator. By setting $n = N = 5$, the ma-

trices $U^{\mathrm{d}}, V^{\mathrm{d}}, W^{\mathrm{d}}$ represent the flow field variables that might have been obtained via a hypothetical CFD grid with 5 nodes simulated for 5 time iterations. The matrix u^{d} and u^{dq} are created next. Every time algorithm 1 is called to create the matrices u^{d} and

Algorithm 1: Function to create a random data set using the dimensions n and N.

input : Number of nodes n, number of snapshots N.
output: Snapshot matrix u^{d}, quaternion snapshot matrix u^{dq}.

Function `Create Data(`n, N`)`**:**

- **while** *abs(norm(U^d,'Frobenius') - 2.88) < 0.0001* **do**
  - U^{d} = rand(n,N);
- **while** *abs(norm(V^d,'Frobenius') - 2.88) < 0.0001* **do**
  - V^{d} = rand(n,N);
- **while** *abs(norm(W^d,'Frobenius') - 2.88) < 0.0001* **do**
  - W^{d} = rand(n,N);
- u^{d} = [U^{d};V^{d};W^{d}];
- u^{dq} = quaternion(U^{d},V^{d},W^{d});
- **return** u^{d},u^{dq}

u^{dq}, algorithm 1 guarantees that the data matrices are different and that they are balanced ($\|U^{\mathrm{d}}\|_{\mathrm{F}} \approx \|V^{\mathrm{d}}\|_{\mathrm{F}} \approx \|W^{\mathrm{d}}\|_{\mathrm{F}} \approx 2.88$). Algorithm 1 also guarantees that the one thousand data sets generated for this experiment have very similar norms.

In order to compare the performance of the POD and the QPOD methods fairly, comparisons should only be done to matrices that have very similar norms to each other, this avoids the effect of scaling as seen in the following relation

$$\mathrm{SVD}(k \times u^{\mathrm{d}}) = k \times \mathrm{SVD}(u^{\mathrm{d}}), \tag{4.1}$$

$$\mathrm{QSVD}(k \times u^{\mathrm{dq}}) = k \times \mathrm{QSVD}(u^{\mathrm{dq}}), \tag{4.2}$$

where $k \in \mathbb{R}$ is a scaling factor.

Three measures are taken to ensure that any problems that may arise due to the norm of the randomly generated data is circumvented:

1. The normalized distance metric η_n is used instead of η.

2. The performance measures ϵ_m^{r} and ϵ_m^{q} are used instead of J_m^{r} and J_m^{q}.

3. It was observed that for a five by five randomly generated matrix using the `rand` function in `MATLAB`, the expected norm of such a matrix is around 2.88. Therefore, the implementation shown in pseudo-code 1 is a fast way to generate different data sets but with similar norms.

In this fashion, any difference in results between the POD and the QPOD is narrowed down to the difference between the methods.

After the data matrices are created, the POD methods and the QPOD method are applied to the data. This is done one thousand times, as outlined in pseudo-code 2. The distance metric η_{n} and the performance measures $\epsilon_m^{\mathrm{r}}, \epsilon_m^{\mathrm{q}}$ and ϵ_m are computed.

Algorithm 2: Perform the POD and QPOD methods to a thousand data sets.

Function `Main`**:**
- **for** $i = 1$ **to** 1000 **do**
  - [u^{d}, u^{dq}] = `Create Data` (5,5);
  - L = norm(u^{d},'Frobenius');
  - [Φ, Σ, A] = SVD (u^{d});
  - [$\Phi^{\mathrm{q}}, \Sigma^{\mathrm{q}}, A^{\mathrm{q}}$] = QSVD($\mathrm{u}^{\mathrm{dq}}$);
  - $\eta(i) = 0$;
  - $\mathrm{u}_{\mathrm{m}}^{\mathrm{d}}$ = zeros($n \times 3$,N);
  - $\mathrm{u}_{\mathrm{m}}^{\mathrm{dq}}$ = quaternion(zeros(n,N),zeros(n,N),zeros(n,N));
  - **for** $m = 1$ **to** 5 **do**
    - $\eta(i) = \eta(i) + \left(\Sigma_q(m,m) - \Sigma_r(m,m)\right)^2$;
    - $\mathrm{u}_{\mathrm{m}}^{\mathrm{d}} = \mathrm{u}_{\mathrm{m}}^{\mathrm{d}} + \Phi(:,m) * \Sigma(m,m) * A(:,m)'$;
    - $\mathrm{u}_{\mathrm{m}}^{\mathrm{dq}} = \mathrm{u}_{\mathrm{m}}^{\mathrm{dq}} + \Phi^{\mathrm{q}}(:,m) * \Sigma^{\mathrm{q}}(m,m) * A^{\mathrm{q}}(:,m)'$;
    - $\epsilon^{\mathrm{r}}(i,m)$ = norm(u^{d} - $\mathrm{u}_{\mathrm{m}}^{\mathrm{d}}$,'Frobenius')/L*100;
    - $\epsilon^{\mathrm{q}}(i,m)$ = norm(u^{dq} - $\mathrm{u}_{\mathrm{m}}^{\mathrm{dq}}$,'Frobenius')/L*100;
  - $\eta_{\mathrm{n}}(i) = \eta(i)$/(2L);
  - $\epsilon(i,:) = \epsilon^{\mathrm{r}}(i,:) - \epsilon^{\mathrm{q}}(i,:)$;

4.2.2 Sanity Check

In order to make sure that experiment one runs smoothly, a sanity check is performed. The purpose of the sanity check is to verify that:

1. The QPOD library provided by Sangwine [100] is properly installed and functioning as predicted.

2. Given a random data set, the POD and the QPOD methods do in fact result in different decompositions.

Figure 4.1 shows the normalized singular values associated with the POD method and the QPOD method plotted on a logarithmic scale for a random data set generated using the `rand` function in `MATLAB`. The results are for $U^{\mathrm{d}}, V^{\mathrm{d}}, W^{\mathrm{d}} \in \mathbb{R}^{50\times 20}$.

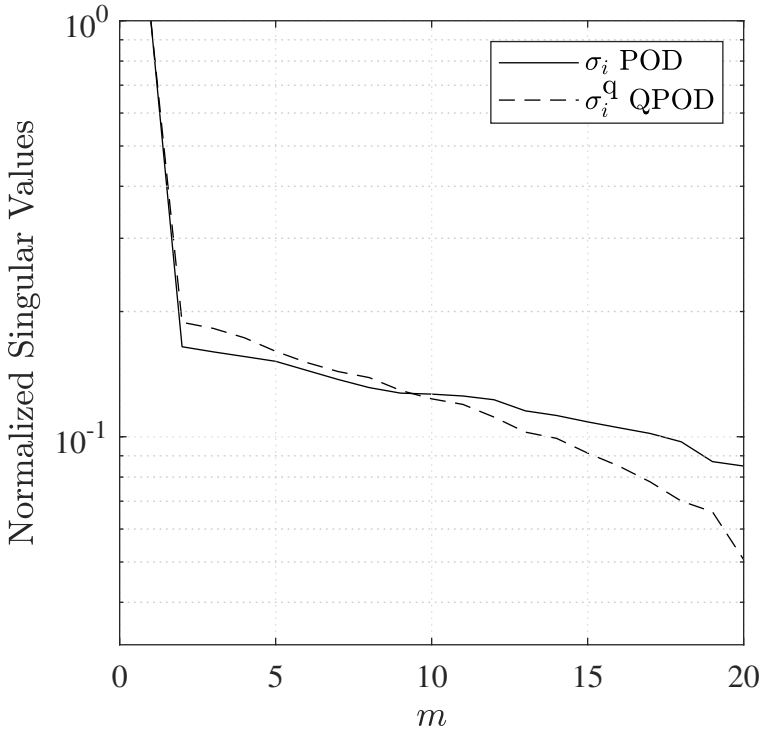


Figure 4.1: Normalized singular values a random data set demonstrating some of the differences between the POD method and the QPOD method. The quaternion singular values $\{\sigma_i^{\mathrm{q}}\}_{i=1}^{20}$ have a steeper rate of decent.

4.3 Experiment Two

Section 3.4.1 outlines the reasoning behind the formulation of the distance metric η. The distance metric only quantifies the extent of difference between the POD modes and the QPOD modes, it does not compare the performance of the modes. The normalized distance metric, equation 3.81, normalizes the distance metric, its expression is repeated here as follows

$$\eta_{\text{n}} = 1 - \frac{1}{L^2} \sum_i \sigma_i^{\text{q}} \sigma_i, \tag{4.3}$$

where L is the norm of the data set. By inspection, it is clear that data sets generated with a small $\sum_i \sigma_i^q \sigma_i$ have a large η_{n} while data sets with large $\sum_i \sigma_i^q \sigma_i$ have a small η_{n}. This implies that, by careful manipulation of the matrices $U^{\text{d}}, V^{\text{d}}, W^{\text{d}} \in \mathbb{R}^{n \times N}$, it is possible to generate data sets not randomly, but that maximizes or minimizes the distance metric, i.e. the differences between the POD modes and the QPOD modes.

In experiment one, the POD and the QPOD methods were applied to randomly generated data sets. This experiment, applies the POD and QPOD methods to data sets obtained in a controlled manner to test the main hypothesis 1. The first data set is carefully generated such that the differences between its coherent structures and its quaternion coherent structures are exaggerated (η maximized). The second data set is also carefully generated such that the differences between its coherent structures and its quaternion coherent structures are minimized (η minimized). This was made possible using `MATLAB`'s optimization routine `fmincon`. The details are outlined next.

4.3.1 Experiment Details

In this experiment, two data sets are generated. A data set consisting of the matrices $U^{\text{d}}_{\text{max}}, V^{\text{d}}_{\text{max}}, W^{\text{d}}_{\text{max}} \in \mathbb{R}^{5 \times 5}$, and a data set consisting of the matrices $U^{\text{d}}_{\text{min}}, V^{\text{d}}_{\text{min}}, W^{\text{d}}_{\text{min}} \in \mathbb{R}^{5 \times 5}$. The following matrices $u^{\text{d}}_{\text{max}}, u^{\text{d}}_{\text{min}} \in \mathbb{R}^{15 \times 5}$ and $u^{\text{dq}}_{\text{max}}, u^{\text{dq}}_{\text{min}} \in \mathbb{H}^{5 \times 5}$ associated with each data

set are defined as follows

$$u^{\mathrm{d}}_{\mathrm{max}} = \begin{bmatrix} U^{\mathrm{d}}_{\mathrm{max}} \\ V^{\mathrm{d}}_{\mathrm{max}} \\ W^{\mathrm{d}}_{\mathrm{max}} \end{bmatrix}, \qquad u^{\mathrm{d}}_{\mathrm{min}} = \begin{bmatrix} U^{\mathrm{d}}_{\mathrm{min}} \\ V^{\mathrm{d}}_{\mathrm{min}} \\ W^{\mathrm{d}}_{\mathrm{min}} \end{bmatrix},$$

$$u^{\mathrm{dq}}_{\mathrm{max}} = U^{\mathrm{d}}_{\mathrm{max}}\boldsymbol{i} + V^{\mathrm{d}}_{\mathrm{max}}\boldsymbol{j} + W^{\mathrm{d}}_{\mathrm{max}}\boldsymbol{k}, \qquad u^{\mathrm{dq}}_{\mathrm{min}} = U^{\mathrm{d}}_{\mathrm{min}}\boldsymbol{i} + V^{\mathrm{d}}_{\mathrm{min}}\boldsymbol{j} + W^{\mathrm{d}}_{\mathrm{min}}\boldsymbol{k}.$$

For the first data set, the $\sum_i \sigma_i^{\mathrm{dq}}\sigma_i$ is minimized, maximizing η. For the second data set, the $\sum_i \sigma_i^{\mathrm{dq}}\sigma_i$ is maximized, minimizing η. In order to generate the mentioned matrices, the `fmincon` optimization routine in `MATLAB` was used with constraint on the norm of the data set, i.e. $\|u^{\mathrm{d}}\|_{\mathrm{F}} = \|u^{\mathrm{dq}}\|_{\mathrm{F}} = 1$. In this manner, the results between the two data sets can be compared without worrying about any scaling issues. In other terms, the constraint enforces the following

$$\sum_{i=1}^{5} (\sigma_i)^2 = \sum_{i=1}^{5} (\sigma_i^{\mathrm{q}})^2 = 1,$$

while the optimization routine attempts to find matrices U, V, and W that maximize or minimize $\sum_i \sigma_i^q \sigma_i$.

The POD method was applied to matrices $u^{\mathrm{d}}_{\mathrm{max}}$ and $u^{\mathrm{d}}_{\mathrm{min}}$ while the QPOD method was applied to matrices $u^{\mathrm{dq}}_{\mathrm{max}}$ and $u^{\mathrm{dq}}_{\mathrm{min}}$. This results in:

1. Two sets of POD singular values, $\{\sigma_i(u^{\mathrm{d}}_{\mathrm{max}})\}_{i=1}^5$ and $\{\sigma_i(u^{\mathrm{d}}_{\mathrm{min}})\}_{i=1}^5$.
2. Two sets of POD modes associated with the matrices $u^{\mathrm{d}}_{\mathrm{max}}$ and $u^{\mathrm{d}}_{\mathrm{min}}$.
3. Two sets of QPOD singular values, $\{\sigma_i^q(u^{\mathrm{dq}}_{\mathrm{max}})\}_{i=1}^5$ and $\{\sigma_i^q(u^{\mathrm{dq}}_{\mathrm{min}})\}_{i=1}^5$.
4. Two sets of QPOD modes associated with the matrices $u^{\mathrm{dq}}_{\mathrm{max}}$ and $u^{\mathrm{dq}}_{\mathrm{min}}$.

The performance measures for each data set were then computed, particularly ϵ_m^r, ϵ_m^q, and their difference ϵ_m.

4.4 Experiment Three

In this experiment, the POD and QPOD direct methods are applied to a large experimental data set obtained from the NASA Langley research facility. The data set obtained is from a series of experiments dedicated to understanding how excrescences (surface imperfections) affect laminar flow and cause the flow to transition. The transition process over excrescences is complex, and in order to create more accurate, and better predictive models of the transition, it is important to understand the underlying mechanisms that cause the transition. Ultimately, these models would help engineers to achieve better laminar flow control in the presence of rivets, bolts, gaps, paint, etc. Unlike experiments one and two, in this experiment the POD and QPOD methods were applied to a set of measurements of a fluid system.

4.4.1 Experiment Details

Eppink [111, 112] performed experiments to investigate the effects of a forward facing step (FFS) on stationary cross-flow growth by using stereo particle image velocimetry (SPIV) measurements. The experiment was carried out at the 2-Foot by 3-Foot Low Speed Boundary-Layer Channel at the NASA Langley Research Center. The test section of the tunnel is 0.61 meters high, 0.91 meters wide, and 6.1 meters long. The tunnel is a low-disturbance facility for conducting transition experiments since in an empty test section, the free-stream turbulence was measured to be less than 0.06% for the entire speed range, and less than 0.05% for the speed of $26.5\,\mathrm{m\,s^{-1}}$.

The model is 0.91 meters wide, 2.54 meters long on the longest edge, and $12.7\,\mathrm{mm}$ thick flat plat. It consists of a 0.41 meter long leading edge piece, swept at $30°$, a larger piece downstream. Figure 4.2 shows a schematic of the experimental setup. The leading edge piece was polished to a surface finish of $0.2\,\mu\mathrm{m}$ rms, and the larger downstream piece was polished to a surface finish of $0.4\,\mu\mathrm{m}$ rms.

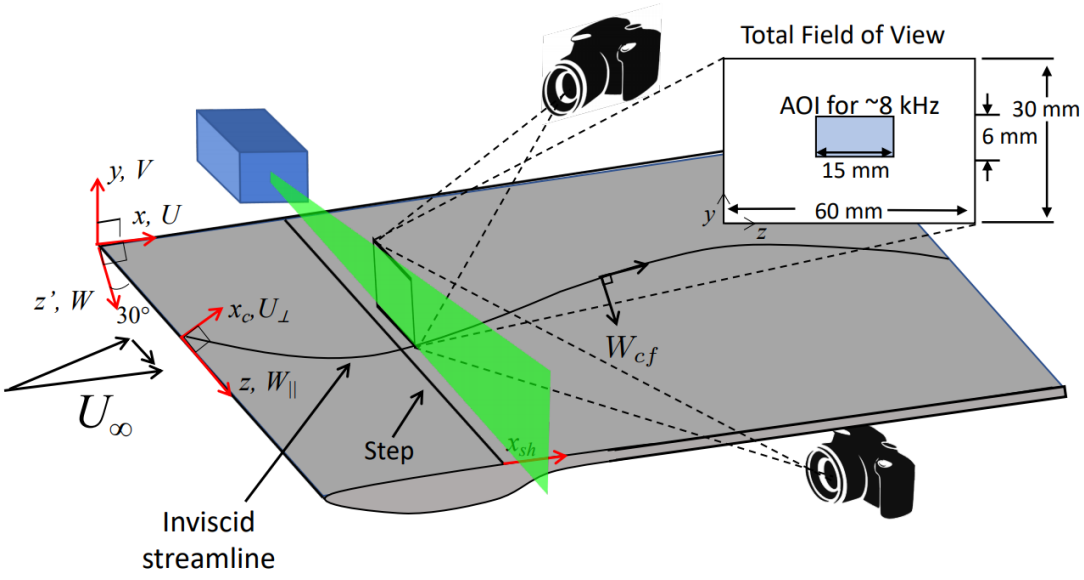


Figure 4.2: Experimental setup to measure the flow over a forward facing step at six different stations.

To instigate stationary stream-wise cross-flow growth, a stream-wise pressure gradient is needed, hence a three dimensional pressure body along the ceiling was designed. This also simulates an infinite swept-wing flow within a mid span measurement region of width 0.3 meters. The ceiling liner was fabricated out of hard foam using computer-controlled milling machining such that the C_p contours are parallel with the leading edge within the measurement region.

The free-stream velocity was set to $26.5\,\mathrm{m\,s^{-1}}$ ($Re' = 1.69 \times 10^6\,\mathrm{m^{-1}}$) throughout the whole experiment and data collection process. A single leading-edge roughness configuration consisting of discrete roughness element (DREs) with a diameter of $4.4\,\mathrm{mm}$ was used. The DREs were configured with a span-wise spacing, λ_z, of $11\,\mathrm{mm}$ and an approximate thickness of $20\,\mu\mathrm{m}$. More details on the experimental setup can be found in the reference by Eppink [113].

A high-speed double-pulsed ND:YLF laser provided the laser sheet for the PIV measurements. The laser was setup such that the laser sheet is parallel to the leading edge and the forward-facing step By placing two high-speed 4-megapixel cameras downstream of the

step, time-resolved PIV (TRIPV) measurements were acquired. Using the apparatus in the configuration described and shown in Figure 4.2, the total possible measurement area came out to $60\,\mathrm{mm} \times 30\,\mathrm{mm}$, which was reduced for the majority of measurements to $15\,\mathrm{mm} \times 6\,\mathrm{mm}$ at an acquisition rate of $8\,\mathrm{kHz}$. The cameras and laser were all mounted on a traversing system, which allows for measurements at multiple locations. The data is acquired at multiple stations, starting at or shortly downstream of the step, moving downstream at approximately $3\,\mathrm{mm}$ increments. 6000 image pairs were acquired at each station.

4.4.2 Data Processing

Using the experimental setup described, the data was carefully collected and stored. Six data sets were collected at different downstream locations. The data sets were labeled according to their station number, which increases further downstream from the step. The data for each station was imported into `MATLAB`. For each station, the data collected reflects the u, v, w velocities of the flow with 129×28 data points per snapshot ($n = 3,612$) and a total of 6,000 snapshots ($N = 6,000$). Hence, for each data set, the matrices $u^{\mathrm{d}} \in \mathbb{R}^{3n \times N}$ and $u^{\mathrm{dq}} \in \mathbb{H}^{n \times N}$ were created. The POD and the QPOD method was then applied to each matrix u^{d} and u^{dq} respectively; the normalized distance metric and the performance measures were computed.

Typically, in POD applications, the number of data points n per snapshot is much larger than the number of snapshots, N. However, in this case, the number of snapshots is larger than the number of data points per snapshot, i.e. $N > n$. Therefore, it is more numerically efficient to apply the direct implementations of the POD and the QPOD methods.

4.5 Experiment Four

As mentioned before, the quaternion POD method becomes computationally infeasible when the dimensionality of the data set becomes large. This is typically the case for average-sized CFD grids, where the number of nodes is in the order of millions. The

proposed quaternion snapshot method is an implementation of the quaternion POD that enables scientist and engineers to perform the QPOD on data sets where the number of nodes is much larger than the number of snapshots. In order to demonstrate a typical modern day situation where the QPOD method would be used, a CFD grid for a three-dimensional cylinder was chosen. The flow around a 3D cylinder is a canonical problem in the fluid flow community that has been used to demonstrate a variety of fluid flow phenomenons and data analysis techniques. Even though the 3D cylinder has a simple geometry, the flow around it can be very rich. The 3D cylinder exhibits a variety of parameter-dependent fluid flow phenomena that has been verified mathematically, experimentally, and numerically, elevating it to a benchmark problem.

The most notable phenomena is the Reynolds number dependent bifurcations that occur as the flow transitions from laminar to turbulent. At low Reynolds numbers, below 46, the flow is asymptotically stable, two dimensional, and tends towards a fixed point [114]. An example of how the steady state solution looks like for Reynolds number 35 is shown in figure 5.15. As the Reynolds number increases beyond the critical Reynolds number $Re_c \approx 46$, the asymptotically stable fixed point mentioned becomes unstable and the solution tends towards a stable, periodic, two dimensional attractor due to a phenomenon known as Hopf bifurcation. This forms a periodic, two dimensional flow, widely known as the Benard-Von Karmen vortex Street [115]. Figure 5.17 shows a snapshot of the Benard-Von Karmen vortex street at Reynolds number 120. The flow is two dimensional due to span-wise translational symmetry along the cylinder. As the Reynolds number increases beyond 188, a supercritical Hopf bifurcation occurs due to span wise instability. The span-wise instability develops, causing the flow behind the cylinder to become completely three dimensional and periodic [116, 117, 118]. A typical feature of this flow is the development of stream-wise vorticities.

As the Reynolds number keeps increasing, the flow around the three dimensional cylinder continues to bifurcate in a sequence of period doubling that ultimately transitions to

chaotic flow observing the universal scaling laws of nonlinear chaotic dynamical systems [119, 120]. The cascade of period doubling bifurcations has been experimentally documented [117], and numerically investigated through direct numerical simulations of the Navier-Stokes equations [121, 120, 114, 122, 123]. While most experimental and numerical investigations were done for an unconfined cylinder, Kanaris et al. [124] have numerically investigated the confined three dimensional cylinder case.

The simple geometry, and the richness of parameter dependent phenomena, has made the three-dimensional cylinder appealing to many scientist and engineers. Over the years, the three-dimensional cylinder has become a benchmark problem in the reduced-order modeling community. Model order reduction algorithms seek to capture the rich dynamics of the wake behind the three dimensional cylinder using the lowest order models possible. The accuracy, stability, and adaptability are a few properties that have been investigated for the resulting reduced order models of the three dimensional cylinder. The POD had a great influence in this field and until this day remains a standard tool in creating reduced-order models of the three dimensional cylinder.

In 1991, Deane et al. [125], used the POD to create a reduced order model for the flow around a two dimensional cylinder. One of the first models to create a reduced order model of the three dimensional cylinder by utilizing POD modes and Galerkin projection methods was created by Noack and Eckelmann [126] in 1994. Many other contributions have been added to the creation of low dimensional models of the wake behind the three dimensional cylinder [121, 120, 127]. Low dimensional models of the three dimensional cylinder have also been utilized in flow control [55, 56]. The literature on this topic is vast and outside the scope of this thesis.

In order to generate a computational model of the three dimensional cylinder that captures the mentioned phenomena, a CFD grid with a large number of nodes was created. The model is then run on NASA's super computer, known as the K-Cluster, at different Reynolds numbers. The resulting data sets were then carefully prepared; the POD method

and the QPOD method were applied.

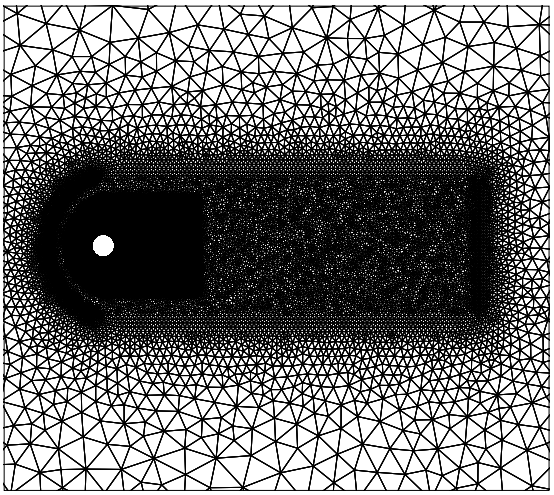
This experiment applies both the POD and the QPOD to the flow around a three dimensional cylinder at different Reynolds numbers. The snapshot implementation was used to implement the POD method, and the quaternion snapshot implementation was used to implement QPOD method. The results of this experiment will show whether the QPOD is scalable, compare the performance of the POD and the QPOD methods, and uncover any parameter dependence of each method at different Reynolds numbers. The aim of this experiment is to test hypotheses 1 and 2.

4.5.1 Grid Generation

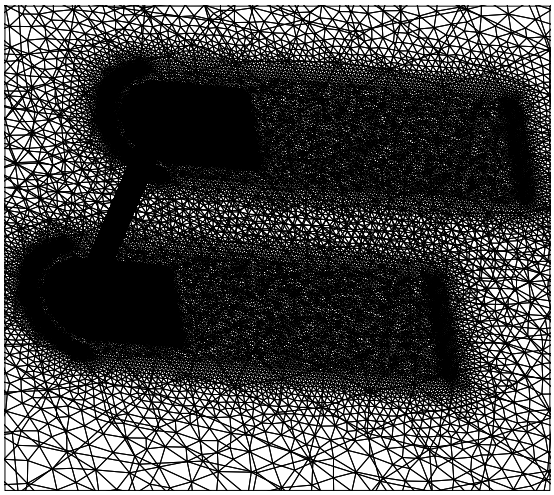
A two dimensional, unstructured grid is first generated in pointwise with 18,775 nodes, as seen in figure 4.3a. The mesh refinement is higher around the cylinder to better capture the flow dynamics in the vicinity of the cylinder. The XZ axes are centered at the center of the cylinder with non-dimensional radius of 1. The X-axis is aligned with the oncoming flow and the Z-axis is perpendicular to the free stream velocity. The grid is then extruded in the Y direction 120 times with a spacing of 0.1 resulting in a three dimensional mesh with 121 layers, a cylinder with span-wise non-dimensional length of 12, and 2,271,775 nodes. The final mesh can be seen in figure 4.3b.

4.5.2 Numerical Method, Boundary Conditions and Simulations

The generated grid is then used with FUN3D, a flow solver, to simulate the flow at different Reynolds numbers. FUN3D [128] is an unstructured node-centered finite-volume solver developed in 1994 and continues to undergo development by NASA. Details of FUN3D including manuals, papers and supporting materials can be found on its official web page [129]. The governing equations used are the viscous incompressible Navier-Stokes equations and the incompressible solver is based on the method of artificial compressibility [130]. FUN3D uses an optimized second-order backwards difference formula-



(a) Side view of the CFD grid. View aligned with the Y-axis showing the XZ plane.



(b) View showing three dimensional geometry of the CFD grid.

Figure 4.3: CFD mesh

tion [131] with dual time-stepping [132]. The non dimensional time step is set to 0.01 with 30 sub-iterations used to converge the dual time-stepping residual.

Viscous and no-slip boundary conditions are imposed on the cylinder with an external flow over the cylinder with span-wise periodicity enforced. Five simulations are ran at different Reynolds numbers. The Reynolds numbers are chosen at different flow regimes in order to capture some of the Reynolds number dependent dynamics described by Williamson [117]. The specific Reynolds numbers chosen are 35, 120, 240, 300, and 600.

4.5.3 Data Processing

After the simulations are ran, the velocity flow field variables are imported into `MATLAB`. Since the dimensionality of the problem is large, applying the POD and the QPOD can be computationally demanding. In this case, the number of nodes n is equal to 2,271,775, while the number of snapshots, N, is 1000 at most. This makes the matrix $u^{\mathrm{d}} \in \mathbb{R}^{3n \times N}$ and $u^{\mathrm{dq}} \in \mathbb{H}^{n \times N}$.

Computing the POD modes via the direct method involves performing the eigenvalue

decomposition on $u^{\mathrm{d}}(u^{\mathrm{d}})^{\top}$, which now is a much bigger matrix with $3n$ rows and $3n$ columns. Similarly, computing the QPOD modes via the direct method involves performing the eigenvalue decomposition on $u^{\mathrm{dq}}(u^{\mathrm{dq}})^{\top}$ which has n rows and n columns. $u^{\mathrm{dq}}(u^{\mathrm{dq}})^{\star} \in \mathbb{H}^{n\times n}$.

Hence, the snapshot POD and the snapshot QPOD implementations are used instead. After the POD and QPOD are applied to the data, the results are stored and compared.

4.5.4 Sanity Check

Before using the snapshot implementation of the QPOD method on a large data set, the quaternion snapshot implementation is validated on a small data set. Figure 4.4 plots the quaternion singular values obtained using:

1. Quaternion direct implementation

2. Quaternion snapshot implementation

where the QPOD method is applied to a quaternion matrix $u^{\mathrm{dq}} \in \mathbb{H}^{5\times 5}$. The components of u^{dq} are can be found in the appendix. Upon comparison, the quaternion singular values obtained using the quaternion direct implementation are found to be exactly the same as the quaternion singular values obtained using the quaternion snapshot implementation, i.e. $\sigma_i^{\text{Direct}} = \sigma_i^{\text{Snap}}$, $\forall i = 1, \ldots, 5$. The quaternion singular values can be seen in figure 4.4 which plots $\{\sigma_i^{\text{Direct}}\}_{i=1}^{5}$ and $\{\sigma_i^{\text{Snap}}\}_{i=1}^{5}$.

However, upon inspecting the modes of each implementation, it is clear that the modes are not the same, even though the singular values of both implementations are the same. The first two modes, associated with the first two singular values, of each implementation

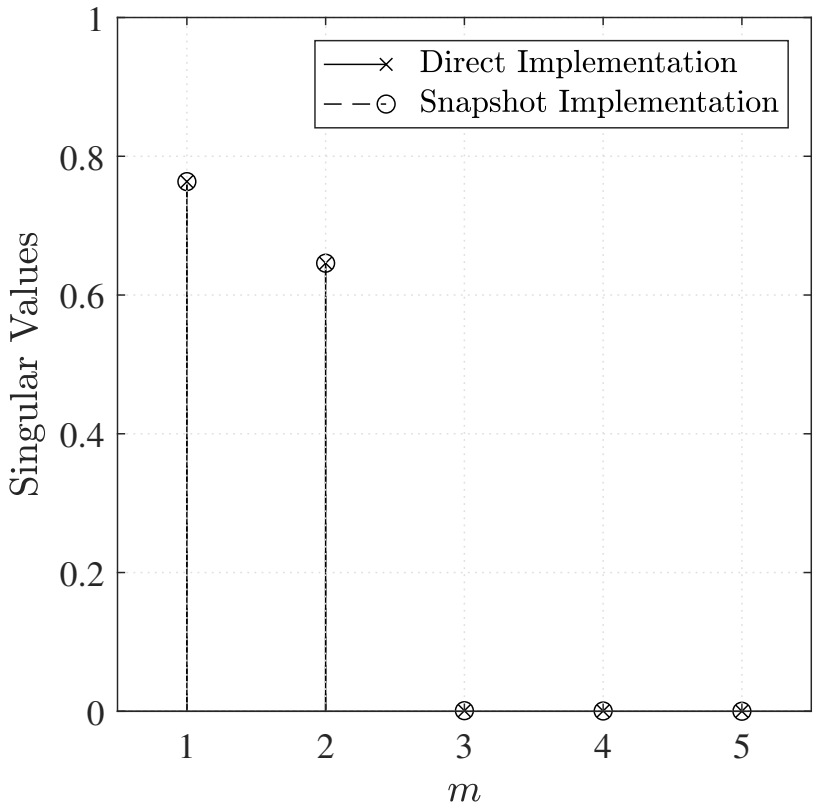


Figure 4.4: Plot shows the quaternion singular values of $u^{\mathrm{dq}}u^{\mathrm{dq}\star}$ and $u^{\mathrm{dq}\star}u^{\mathrm{dq}}$ in perfect agreement.

are listed below

$$\phi_1^{\text{Direct}} = \begin{bmatrix} 0.09 + 0.06\boldsymbol{i} - 0.10\boldsymbol{j} - 0.18\boldsymbol{k} \\ 0.12 + 0.18\boldsymbol{i} - 0.39\boldsymbol{j} - 0.33\boldsymbol{k} \\ -0.02 + 0.06\boldsymbol{i} - 0.16\boldsymbol{j} - 0.02\boldsymbol{k} \\ -0.08 - 0.37\boldsymbol{i} + 0.50\boldsymbol{j} + 0.41\boldsymbol{k} \\ 0.10 + 0.04\boldsymbol{i} + 0.05\boldsymbol{j} - 0.18\boldsymbol{k} \end{bmatrix}, \quad \phi_2^{\text{Direct}} = \begin{bmatrix} 0.06 - 0.17\boldsymbol{i} - 0.24\boldsymbol{j} - 0.15\boldsymbol{k} \\ 0.09 - 0.27\boldsymbol{i} + 0.06\boldsymbol{j} - 0.34\boldsymbol{k} \\ -0.02 + 0.02\boldsymbol{i} + 0.34\boldsymbol{j} - 0.06\boldsymbol{k} \\ -0.06 - 0.31\boldsymbol{i} - 0.05\boldsymbol{j} - 0.25\boldsymbol{k} \\ 0.07 - 0.08\boldsymbol{i} - 0.62\boldsymbol{j} + 0.05\boldsymbol{k} \end{bmatrix}$$

$$\phi_1^{\text{Snap}} = \begin{bmatrix} 0.00 + 0.17\boldsymbol{i} + 0.01\boldsymbol{j} + 0.42\boldsymbol{k} \\ 0.00 - 0.24\boldsymbol{i} + 0.38\boldsymbol{j} + 0.38\boldsymbol{k} \\ 0.01 - 0.02\boldsymbol{i} + 0.25\boldsymbol{j} - 0.25\boldsymbol{k} \\ -0.01 - 0.15\boldsymbol{i} + 0.05\boldsymbol{j} - 0.19\boldsymbol{k} \\ -0.00 - 0.05\boldsymbol{i} - 0.31\boldsymbol{j} - 0.41\boldsymbol{k} \end{bmatrix}, \quad \phi_2^{\text{Snap}} = \begin{bmatrix} -0.02 - 0.29\boldsymbol{i} - 0.12\boldsymbol{j} - 0.20\boldsymbol{k} \\ 0.06 - 0.15\boldsymbol{i} + 0.02\boldsymbol{j} + 0.24\boldsymbol{k} \\ 0.09 - 0.22\boldsymbol{i} + 0.40\boldsymbol{j} - 0.29\boldsymbol{k} \\ -0.08 + 0.54\boldsymbol{i} - 0.17\boldsymbol{j} + 0.02\boldsymbol{k} \\ 0.05 - 0.24\boldsymbol{i} - 0.08\boldsymbol{j} + 0.29\boldsymbol{k} \end{bmatrix}$$

By examining the entries of the vectors, it is clear that the resulting modes are not the same ($\phi_i^{\text{Direct}} = \phi_i^{\text{Snap}}$, $i = 1, 2$), even though the eigenvalues are the same ($\lambda_i^{\text{Direct}} = \lambda_i^{\text{Snap}}$, $i = 1, 2$). This results is unexpected as both implementations should yield the same set of modes.

In order to verify which implementation produces the correct QPOD modes, recall that the modes are actually eigenvectors that satisfy the following relationship

$$u^{\text{dq}} u^{\text{dq}^\star} \phi_i^{\text{q}} = \phi_i^{\text{q}} \lambda_i^{\text{q}}. \tag{4.4}$$

Therefore, the eigenvalue-eigenvector relationship mentioned above is tested for the modes obtained via the direct implementation and for the modes obtained via the quaternion snapshot implementation. Using `MATLAB`, both of the following equations were satisfied,

$$\left(u^{\text{dq}} u^{\text{dq}^\star}\right) \phi_i^{\text{Direct}} - \phi_i^{\text{Direct}} \lambda_i^{\text{q}} = 0, \qquad \forall i = 1 \ldots 5, \tag{4.5}$$

$$\left(u^{\text{dq}} u^{\text{dq}^\star}\right) \phi_i^{\text{Snap}} - \phi_i^{\text{Snap}} \lambda_i^{\text{q}} = 0, \qquad \forall i = 1 \ldots 5. \tag{4.6}$$

Therefore, both sets of vectors $\{\phi_i^{\text{Direct}}\}_{i=1}^5$ and $\{\phi_i^{\text{Snap}}\}_{i=1}^5$ satisfy equation 4.4. Hence, by definition $\{\phi_i^{\text{Direct}}\}_{i=1}^5$ and $\{\phi_i^{\text{Snap}}\}_{i=1}^5$ are QPOD modes of u^{dq}, which seems puzzling.

The answer to the puzzle can be found in quaternion matrix algebra. Equation 4.5 shows that $(\lambda_i^{\text{q}}, \phi_i^{\text{Direct}})$ is an eigenpair of $u^{\text{dq}}u^{\text{dq}^\star}$, and equation 4.6 also shows that $(\lambda_i^{\text{q}}, \phi_i^{\text{Snap}})$ is an eigenpair of $u^{\text{dq}}u^{\text{dq}^\star}$. Therefore, both ϕ_i^{Direct} and ϕ_i^{Snap} belong to the same kaleidoscope set $\kappa(\lambda_i^{\text{q}})$, where

$$\kappa(\lambda_i^{\text{q}}) = \big\{q \in \mathbb{H}^n : q = \phi^{\text{q}} w, w \in \mathbb{H}, w \neq 0\big\}. \tag{4.7}$$

Consequently, for quaternion hermitian matrices such as $u^{\text{dq}}u^{\text{dq}^\star}$, the vectors ϕ_i^{Direct} and ϕ_i^{Snap} are both eigenvectors and belong to the same class of eigenvectors. By theorem 4, this means that for all $i = 1 \ldots 5$, there exists a unit quaternion $w_{\text{rot}} \in \mathbb{H}$, where $\|w_{\text{rot}}\| = 1$, such that

$$\phi_i^{\text{Direct}} = \phi_i^{\text{Snap}} w_{\text{rot}}. \tag{4.8}$$

In fact there are infinitely many eigenvectors associated with every eigenvalue λ_i^{q}. As long as the eigenvectors are multiplied by a unit quaternion, the result is also an eigenvector that belongs to the same class of eigenvectors. Since multiplying by a unit quaternion is equivalent to rotation in four dimensional space, this means that all the eigenvector are equivalent up to four dimensional rotations of their entries.

CHAPTER 5
RESULTS AND DISCUSSION

In chapter 2, the aim of this thesis was stated as follows:

Research Objective. *Explore the use of quaternions in the context of modal analysis and reduced-order modeling of three-dimensional fluid systems.*

In chapter 3, the quaternion proper orthogonal decomposition was shown to yield promising results in the fields of color image compression [90] and signal processing [89]. However, little attention has been paid to application of the quaternion proper orthogonal decomposition to three dimensional fluid systems. This motivated the main research question (research question 1) of this thesis

Research Question 1. *How would a quaternion approach to the POD method in the context of modal analysis and reduced-order modeling of three-dimensional fluid systems compare to the traditional POD method?*

In chapter 4, four experiments that compare the POD method and the QPOD method were thoroughly formulated with the purpose of answering research question 1; the experiments were then carefully conducted. However, just like the POD method, the direct implementation of the QPOD method also suffers from the lack of scalability. Hence, the quaternion snapshot implementation was devised as a means to apply the QPOD method to large data sets, such as the ones encountered in experiment four. This led to research question 2 restated here as follows:

Research Question 2. *Is the quaternion proper orthogonal decomposition scalable?*

The present chapter has two sections. The first section presents and discusses the results of these four experiments. The second section discusses some mathematical observations

that can help explain why those results might have occurred.

5.1 Experiments

The theme of this section relies on applying the POD method and the QPOD method to different data sets. In experiment one and two, the data sets ($U^{\mathrm{d}}, V^{\mathrm{d}}, W^{\mathrm{d}} \in \mathbb{R}^{5\times5}$) were generated either randomly or via an optimization routine. The random generation of many but small data sets in experiment one allowed for the comparison between the average performance of the two methods, and helped in revealing important trends.

In experiment three and four, the methods were applied to data sets pertaining to three-dimensional fluid systems. These experiments serve to show case the QPOD method as it applies to fluid systems, and compares the performance between POD method and the QPOD method.

In every one of the four experiments, a data set $U^{\mathrm{d}}, V^{\mathrm{d}}, W^{\mathrm{d}} \in \mathbb{R}^{n\times N}$ was used to create the snapshot matrix $u^{\mathrm{d}} \in \mathbb{R}^{(n\times3)\times N}$ and the quaternion snapshot matrix $u^{\mathrm{dq}} \in \mathbb{H}^{n\times N}$, where n is the number of data points per snapshot, and N is the total number of snapshots. The POD method and the QPOD method were applied to data matrix u^{d} and u^{dq}, respectively. The results of the process described is summarized as follows:

- The traditional POD method applied to the snapshot matrix, u^{d}, resulted in:
  1. POD modes, $\{\phi_i(X)\}_{i=1}^m$, also known as coherent structures.
  2. Singular values associated with each mode, $\{\sigma_i\}_{i=1}^m$.
- The quaternion POD method applied to the quaternion snapshot matrix, u^{dq}, resulted in:
  1. QPOD modes, $\{\phi_i^q(X)\}_{i=1}^m$, defined as quaternion coherent structures.
  2. Quaternion singular values associated with each quaternion mode, $\{\sigma_i^q\}_{i=1}^m$.

By comparing the modes, singular values, and computing the distance metric (η) and performance measures ($\epsilon_m^{\mathrm{r}}, \epsilon_m^{\mathrm{q}}, \epsilon_m$), the POD method is compared to the QPOD method. In addition, experiment four utilized the quaternion snapshot implementation of the QPOD method, which was applied at different Reynolds numbers.

5.1.1 Experiment One

Using `MATLAB`, a thousand U^{d} matrices, a thousand V^{d} matrices, and a thousand W^{d} matrices were generated, as outlined in algorithm 1. The dimension of each generated matrix was chosen to be five by five, $U^{\mathrm{d}}, V^{\mathrm{d}}, W^{\mathrm{d}} \in \mathbb{R}^{5\times 5}$. Such a data set represents a hypothetical flow where the flow field variables would be extracted from a grid with five nodes over the period of five time iterations. Table 5.1 shows that even though the matrices

	mean norm	std norm
U^{d}	2.88	5.80×10^{-5}
V^{d}	2.88	5.80×10^{-5}
W^{d}	2.88	5.85×10^{-5}
u^{d}	4.99	5.82×10^{-5}
u^{dq}	4.99	5.82×10^{-5}

Table 5.1: Statistics on the randomly generated matrices. The first three rows show the statistics associated with the one thousand randomly generated matrices. The first column is the mean of the Frobenius norm's, while the second column represents the standard deviation of the norms.

were randomly generated, careful attention was given to the process of generating them. The norm of the norm of a randomly generated data set is given by

$$L = \|u^{\mathrm{d}}\|_{\mathrm{F}} = \|u^{\mathrm{dq}}\|_{\mathrm{F}}. \tag{5.1}$$

Table 5.1 shows that algorithm 1 consistently generated data sets with very similar norms, $L \approx 4.99$, due to the negligible variability of the norm among the data sets.

Using the data sets generated, the snapshot matrices u^{d} (defined by equation 2.54), and the quaternion snapshot matrices u^{dq} (defined by equation 3.54) were created. The POD

method was applied to every snapshot matrix u^{d}, while the QPOD method was applied to every quaternion snapshot matrix u^{dq}. Therefore, for each and every data set, this resulted in a set of singular values $\{\sigma_i\}_{i=1}^{5}$ and a set quaternion singular values $\{\sigma_i^{\mathrm{q}}\}_{i=1}^{5}$, which were used to compute the normalized distance metric, as given in equation 3.81. The normalized distance metric computed was sorted and plotted. Figure 5.1 shows the result. It is clear

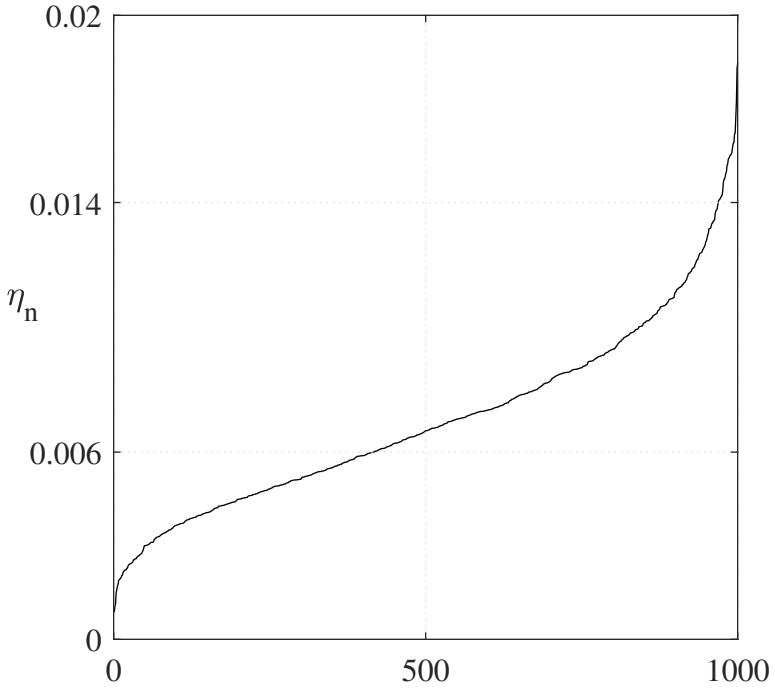


Figure 5.1: Plot shows the normalized distance metric η_{n} (equation 3.81) computed for a thousand randomly generated data sets. The η_{n} values were sorted before they were plotted.

from figure 5.1 that for some data sets, the POD and QPOD methods produce identical singular values $\sigma_i = \sigma_i^{\mathrm{q}}$, $i = 1, \ldots, 5$, and consequently, identical decompositions, $\phi_i = \phi_i^{\mathrm{q}}, i = 1, \ldots, 5$. However, figure 5.1 also shows that the POD and the QPOD generally produce different results.

Observation 14. *The difference in results between the POD method and the QPOD method, as captured by the distance metric η_n, varies across a spectrum.*

The next step is to compare the performance of the two methods. The performance of the two methods is captured via the performance metric ϵ_m^{r} and ϵ_m^{q}. These metrics measure

the ability of the POD modes to approximate the original data set u^{d}, and the ability of the QPOD modes to approximate the same data set in its quaternion representation u^{dq}.

The POD modes and the QPOD modes were used to create approximations of the original data. Four approximations are computed for each data set as follows

$$u_m^{\mathrm{d}} = \sum_{i=1}^{m} \phi_i a_i, \qquad m = 1, \ldots, 4, \tag{5.2}$$

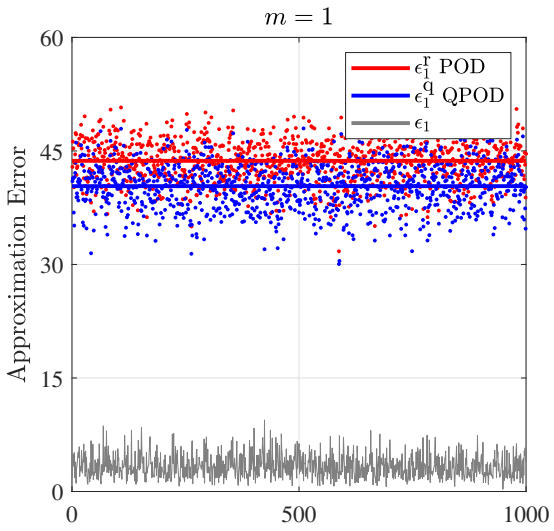
$$u_m^{\mathrm{dq}} = \sum_{i=1}^{m} \phi_i^{\mathrm{q}} a_i^{\mathrm{q}}, \qquad m = 1, \ldots, 4. \tag{5.3}$$

Each approximation was incrementally enriched by adding an extra mode, resulting in four approximations per method per data set. The error associated with each approximation was computed for each data set and plotted. The results can be seen in figure 5.2. The red dots show the performance metric ϵ_m^{r} associated with the POD method, while the blue dots show the error associated with the QPOD, ϵ_m^{r}. The gray line shows the difference between the two errors, $\epsilon_m = \epsilon_m^{\mathrm{r}} - \epsilon_m^{\mathrm{q}}$.

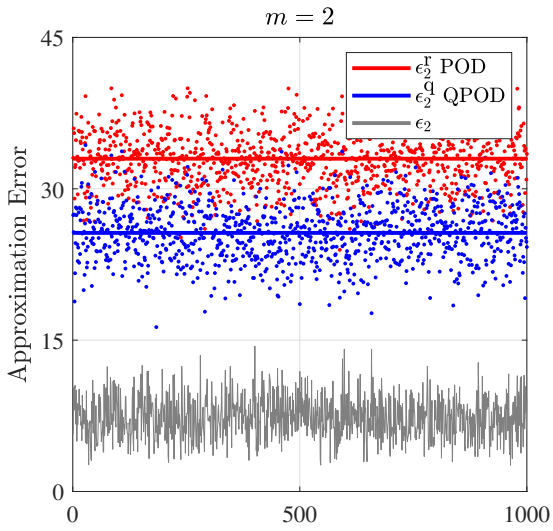
It is evident from the plots that the QPOD always outperforms the POD method, because $\epsilon_m = \epsilon_i^r - \epsilon_i^q \geq 0, i = 1, \ldots, 4$. Moreover, when more modes are added to the approximations u^{d} and u^{dq}, then ϵ_m increases. This implies that the QPOD approximations converge faster to the original data set than the POD approximations. These observations are summarized, as follows

Observation 15. *The performance metric ϵ_m was greater than zero, for all $m = 1, \ldots, 4$, and for all randomly generated data sets. Therefore, without exception, the approximations u_m^{dq} created via the QPOD method performed better than the approximations u_m^d created via the POD method.*

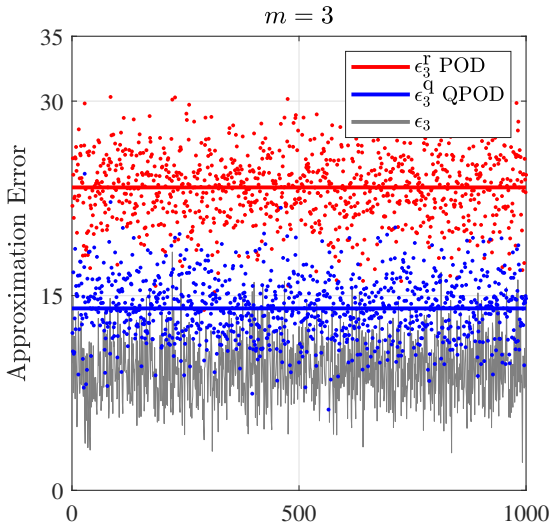
Observation 16. *When more modes were used to create the approximations u_m^d and u_m^{dq}, the performance metric ϵ_m increased. This indicates that the difference between the QPOD approximations u^{dq} and the POD approximations u^d becomes more pronounced when more*



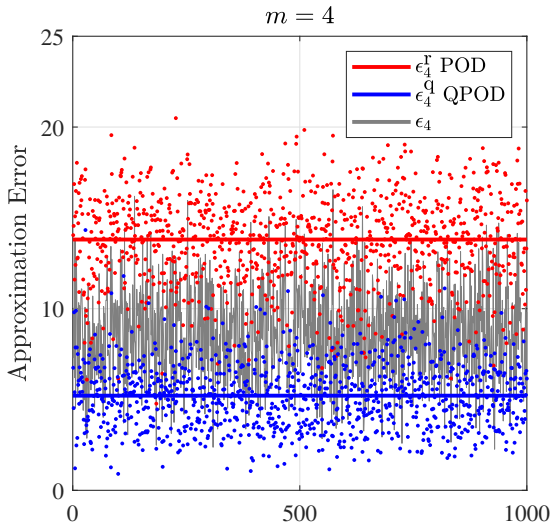
(a) Plot of performance metrics $\epsilon_1^{\mathrm{r}}, \epsilon_1^{\mathrm{q}}$, and ϵ_1, for a thousand randomly generated data sets, which compares the ability of u_1^{d} and u_1^{dq} to approximate the original data set with one mode.



(b) Plot of performance metrics $\epsilon_2^{\mathrm{r}}, \epsilon_2^{\mathrm{q}}$, and ϵ_2, for a thousand randomly generated data sets, which compares the ability of u_2^{d} and u_2^{dq} to approximate the original data set with two modes.



(c) Plot of performance metrics $\epsilon_3^{\mathrm{r}}, \epsilon_3^{\mathrm{q}}$, and ϵ_3, for a thousand randomly generated data sets, which compares the ability of u_3^{d} and u_3^{dq} to approximate the original data set with three modes.



(d) Plot of performance metrics $\epsilon_4^{\mathrm{r}}, \epsilon_4^{\mathrm{q}}$, and ϵ_4, for a thousand randomly generated data sets, which compares the ability of u_4^{d} and u_4^{dq} to approximate the original data set with four modes.

Figure 5.2: Comparison of QPOD performance vs POD performance over one thousand randomly generated data sets. The solid lines represent the average errors. The gray line shows the difference between the two errors, $\epsilon_m = \epsilon_m^r - \epsilon_m^q$. Note that the QPOD always outperforms the POD.

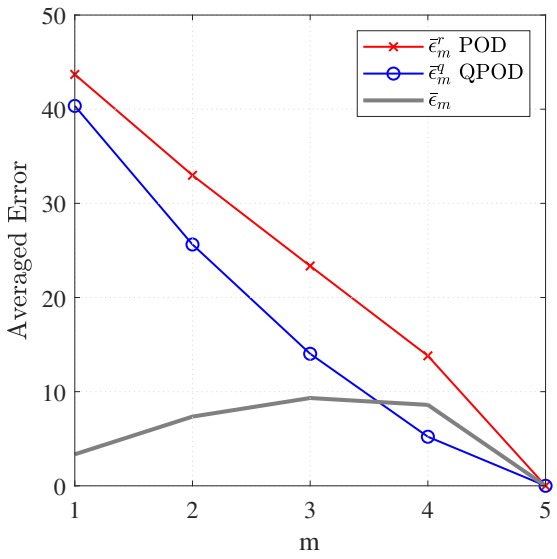


Figure 5.3: The performance metrics ϵ_m^r, ϵ_m^q, and their difference ϵ_m averaged over a thousand data sets plotted against the number of modes m used to create the approximations.

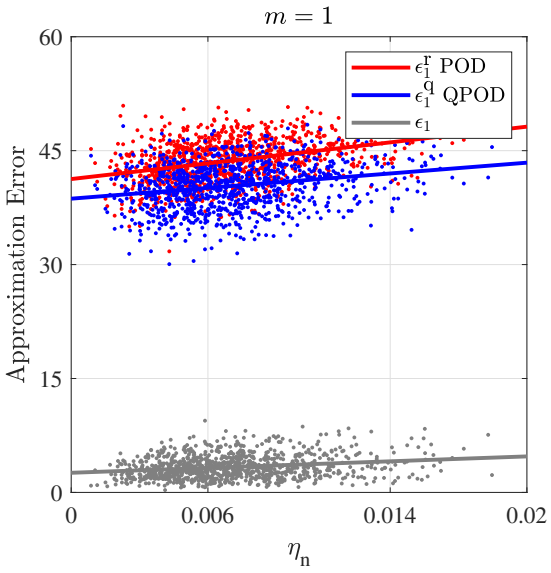
modes are added.

The results of figure 5.2 are summarized in figure 5.3. In figure 5.3, the average error using the POD method ϵ_m^r, the average error using the QPOD method ϵ_m^q, and the average difference of error between the methods ϵ_m, are all plotted against the number of modes m used to create the approximations $u_m^{\mathrm{d}}, u_m^{\mathrm{dq}}$. This indicates that the difference between the average performances between the QPOD method and the POD method becomes more pronounced as the number of modes used to create the approximations increases.

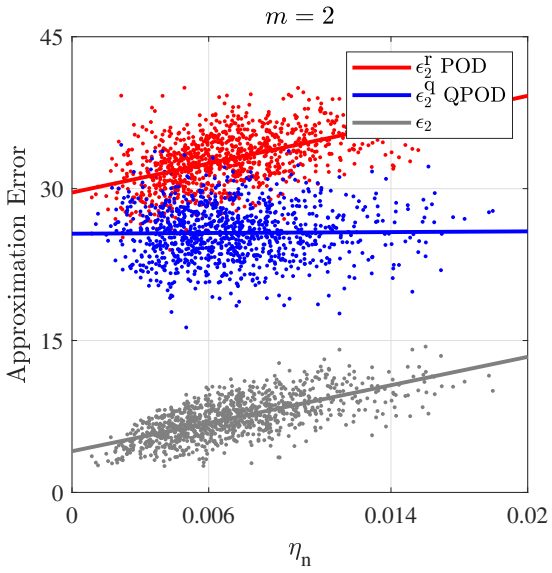
When the performance metrics $\epsilon_m, \epsilon_m^{\mathrm{r}}$, and ϵ_m^{q}, were plotted against the normalized distance metric η_{n}, a strong correlation between those variables was discovered. Figure 5.4, contains four sub-figures Every sub-figure shows how the performance metrics vary with the normalized distance metric, and how those variations change as more modes are used to create the rank-m approximations. The following observations are drawn from figure 5.4.

Observation 17. *When η_n increases, the error ϵ_m^r also increases. This indicates that the POD method performs more poorly for data sets with larger values η_n.*

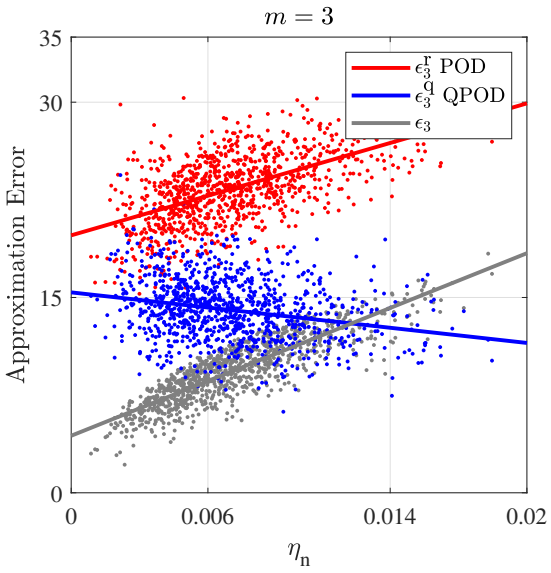
Observation 18. *When more modes are used (larger values of m) to create approximations*



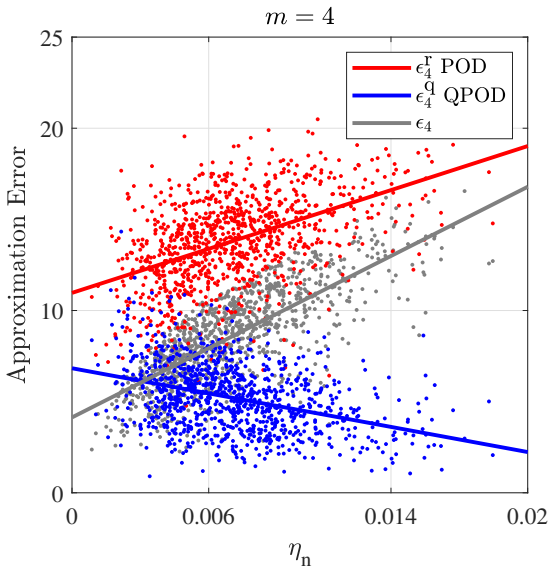
(a) Comparison of POD error vs QPOD error when creating a ROM using the first mode.



(b) Comparison of POD error vs QPOD error when creating a ROM using the first two modes.



(c) Comparison of POD error vs QPOD error when creating a ROM using the three modes.



(d) Comparison of POD error vs QPOD error when creating a ROM using the four modes.

Figure 5.4: Comparison of POD error vs QPOD error as defined by equations 3.74 and 3.76 respectively over one thousand randomly generated data sets. The solid lines represent the best fit line of each respective error. The gray plot shows the difference between the two errors as defined by equation 3.77. Note that the QPOD always outperforms the POD.

u_m^d, the correlation between η_n and the error ϵ_m^r becomes stronger. This indicates that for larger values η_n, the performance of the POD method in approximating the original data set further deteriorates when more modes are used.

Observation 19. *When η_n increases, the error ϵ_m^q decreases. This indicates that the QPOD method performs better for data sets with larger values η_n.*

Observation 20. *As the number of modes used (larger values of m) to create approximations u_m^{dq}, the correlation between the distance metric η_n and the error ϵ_m^q becomes stronger in the negative direction. This indicates that for larger values η_n, the performance of the QPOD method in approximating the original data set is further enhanced when more modes are used.*

Observation 21. *The difference in performance between the POD and the QPOD methods (captured by ϵ_m), becomes more pronounced when:*

1. *Data sets have larger η_n.*

2. *Additional modes are used to create approximations.*

Figure 5.4a shows hardly any correlation between the errors and the distance metric η. This is due to the fact that the first mode is also average mode when the data is not centered, which is the case here. However, when approximations $u_1^{\mathrm{d}}, u_1^{\mathrm{dq}}$ were created using the average mode, without exception, $\epsilon_1^{\mathrm{r}} \geq \epsilon_1^{\mathrm{q}}$. This is surprising as it was verified that both the POD method and the QPOD method produced ϕ_1 and ϕ_1^{q} that are the equivalent to each other (the first mode is the average of the data set when the data set is not centered).

Observation 22. *For all data sets, $\epsilon_1 > 0$; even though $\phi_1 = \phi_1^q$.*

5.1.2 Experiment Two

In this experiment, two hypothetical data sets, data set A and data set B, are generated using the optimization routine `fmincon` available in `MATLAB`. Data sets A and B were generated as follows:

1. Data Set A : matrices $U^{\mathrm{d}}_{\mathrm{max}}, V^{\mathrm{d}}_{\mathrm{max}}$, and $W^{\mathrm{d}}_{\mathrm{max}}$ maximize the distance η.

2. Data Set B : matrices $U^{\mathrm{d}}_{\mathrm{min}}, V^{\mathrm{d}}_{\mathrm{min}}$, and $W^{\mathrm{d}}_{\mathrm{min}}$ minimizes the distance η.

A norm constraint was applied to both data sets such $\|U^{\mathrm{d}}\|^2_{\mathrm{F}} + \|V^{\mathrm{d}}\|^2_{\mathrm{F}} + \|W^{\mathrm{d}}\|^2_{\mathrm{F}} = 1$. This process did not guarantee a globally optimal solution, only a locally optimal one.

With regards to data set A, the exact values of the matrices $U^{\mathrm{d}}_{\mathrm{max}}, V^{\mathrm{d}}_{\mathrm{max}}, W^{\mathrm{d}}_{\mathrm{max}} \in \mathbb{R}^{5\times 5}$ generated can be found in the appendix B. In this data set, the following quantity $\sum_{i=1}^{5} \sigma^{\mathrm{q}}_i \sigma_i$ was found to be equal to 0.6316. Since the norm of the data set was constrained to one, $\|u^{\mathrm{d}}_{\mathrm{max}}\|_{\mathrm{F}} = \|u^{\mathrm{dq}}_{\mathrm{max}}\|_{\mathrm{F}} = 1$, the distance metric was computed as shown in equation 3.80, as follows

$$\begin{aligned} \eta_{\mathrm{max}} &= \|u^{\mathrm{dq}}_{\mathrm{max}}\|^2_{\mathrm{F}} + \|u^{\mathrm{d}}_{\mathrm{max}}\|^2_{\mathrm{F}} - 2\sum_{i=1}^{5} \sigma^{\mathrm{q}}_i(u^{\mathrm{dq}}_{\mathrm{max}})\sigma_i(u^{\mathrm{d}}_{\mathrm{max}}), \\ &= 1 + 1 - 2 \times 0.6316, \\ &= 0.7369, \end{aligned} \tag{5.4}$$

and the normalized distance metric was computed as follows

$$\eta_{\mathrm{n,max}} = \frac{\eta_{\mathrm{max}}}{2} = 0.3684. \tag{5.5}$$

The same process is repeated for the generation of the second data set, $U^{\mathrm{d}}_{\mathrm{min}}, V^{\mathrm{d}}_{\mathrm{min}}, W^{\mathrm{d}}_{\mathrm{min}} \in \mathbb{R}^{5\times 5}$, except this time the optimizer was set to maximize the following quantity $\sum_{i=1}^{5} \sigma^{\mathrm{q}}_i \sigma_i$, which came out to be exactly 1. The exact values of the matrices $U^{\mathrm{d}}_{\mathrm{min}}, V^{\mathrm{d}}_{\mathrm{min}}, W^{\mathrm{d}}_{\mathrm{min}} \in \mathbb{R}^{5\times 5}$

generated can be found in the appendix B. The distance metric was computed as follows

$$\begin{aligned}\eta_{\text{min}} &= \|u^{\text{dq}}_{\text{min}}\|^2_{\text{F}} + \|u^{\text{d}}_{\text{min}}\|^2_{\text{F}} - 2\sum_{i=1}^{5} \sigma^{\text{q}}_i(u^{\text{dq}}_{\text{min}})\sigma_i(u^{\text{d}}_{\text{min}}),\\ &= 1 + 1 - 2 \times 1,\\ &= 0.\end{aligned} \tag{5.6}$$

The normalized distance metric is then equal to

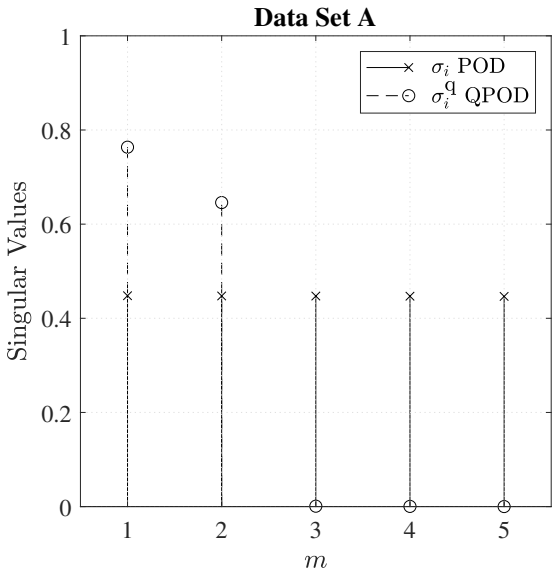
$$\eta_{\text{n,min}} = \frac{\eta_{\text{min}}}{2} = 0. \tag{5.7}$$

Figure 5.5a shows the singular values $\{\sigma_i(u^{\text{d}}_{\text{max}})\}_{i=1}^{5}$ and the quaternion singular values $\{\sigma^{\text{q}}_i(u^{\text{dq}}_{\text{max}})\}_{i=1}^{5}$ associated with data set A, i.e. $U^{\text{d}}_{\text{max}}, V^{\text{d}}_{\text{max}}, W^{\text{d}}_{\text{max}} \in \mathbb{R}^{5\times 5}$ generated such that η is maximized. Figure 5.5b shows the singular values $\{\sigma_i(u^{\text{d}}_{\text{min}})\}_{i=1}^{5}$ and the quaternion singular values $\{\sigma^{\text{q}}_i(u^{\text{dq}}_{\text{min}})\}_{i=1}^{5}$ associated with data set B, i.e. $U^{\text{d}}_{\text{min}}, V^{\text{d}}_{\text{min}}, W^{\text{d}}_{\text{min}} \in \mathbb{R}^{5\times 5}$ generated such that η is minimized. The following observations are made:

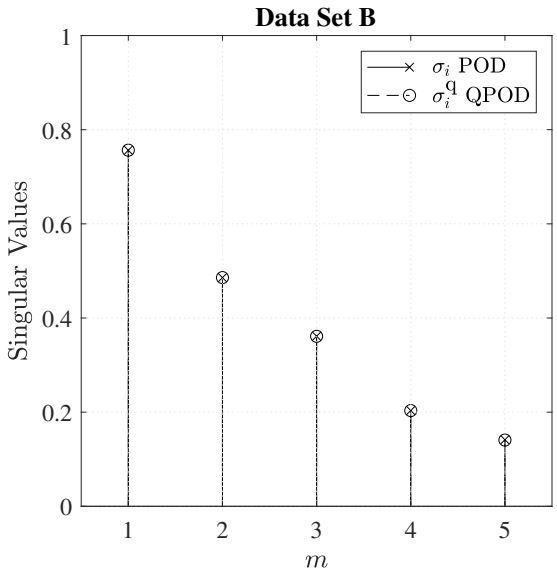
Observation 23. *In data set A, figure 5.5a shows that the rate of decent associated with the quaternion singular values $\{\sigma_i(u^{d}_{max})\}_{i=1}^{5}$ is very steep. Moreover, the last three quaternion singular values are all zero. However, the singular values $\{\sigma_i(u^{d}_{min})\}_{i=1}^{5}$ do not descend at all. This indicates that the POD method was unable to find any pattern in data set A.*

Observation 24. *In data set B, the singular values of both methods exactly coincide, i.e. $\sigma_i = \sigma^q_i$, $\forall i = 1, \ldots, 5$.*

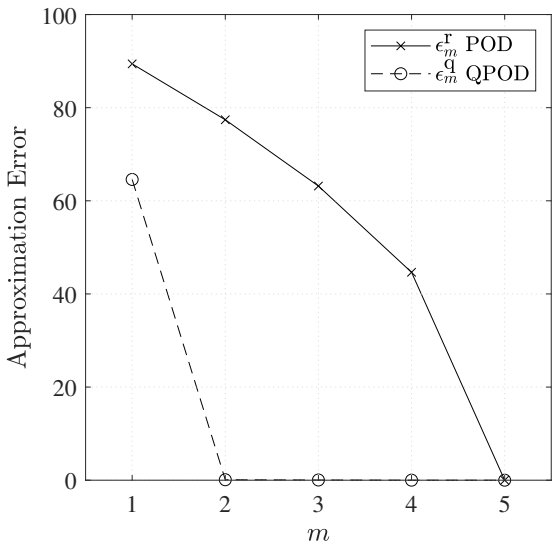
After the POD and QPOD methods were applied to data sets A and B, approximations of each data set were created. Subsequently, the normalized errors associated with each rank-m order approximation was computed. Figure 5.5c reveals an outstanding performance of the QPOD method when compared to the POD method for data set A. However, figure 5.5d reveals that for data set B, both methods had exactly the same performance.



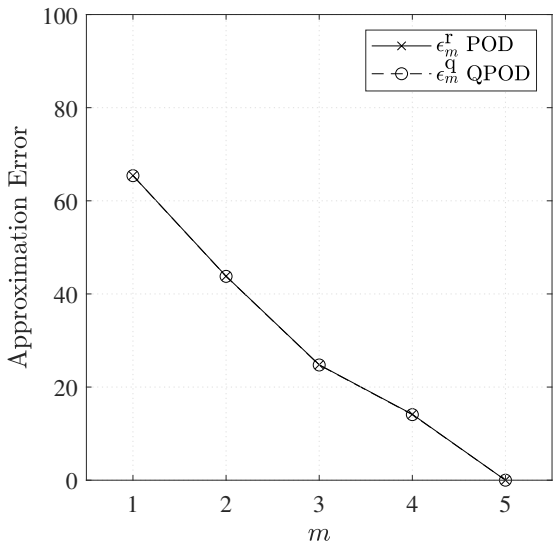
(a) Singular values of the POD method, $\{\sigma_i(u^{\mathrm{d}}_{\max})\}_{i=1}^5$, and the singular values of the QPOD method, $\{\sigma_i^q(u^{\mathrm{dq}}_{\max})\}_{i=1}^5$, shown for data set A where $\eta = \eta_{\max}$.



(b) Singular values of the POD method, $\{\sigma_i(u^{\mathrm{d}}_{\min})\}_{i=1}^5$, and the singular values of the QPOD method, $\{\sigma_i^q(u^{\mathrm{dq}}_{\min})\}_{i=1}^5$, shown for data set B where $\eta = \eta_{\min}$.



(c) Reduced order model errors for the POD method and the QPOD method vs the number of modes used to create the reduced order model. The results shown are for the data set A where the distance metric is maximized.



(d) Reduced order model errors for the POD method and the QPOD method vs the number of modes used to create the reduced order model. The results shown are for the data set B where the distance metric is minimized.

Figure 5.5: The two figures in the left column demonstrate the results for data set A where $\eta = \eta_{\max}$ while the two figure in the right column demonstrate the results for data set B where $\eta = \eta_{\min}$.

The following observations summarize the findings illustrated in figure 5.5:

Observation 25. *For data set A, the error ϵ_m^q is much smaller than ϵ_m^r for all m.*

Observation 26. *For data set A, only two QPOD modes were required to reconstruct the entire data set, while all five POD modes were required to accomplish the same task.*

Observation 27. *For data set B, the POD and the QPOD produce the same singular values and the same modes and hence both methods have equivalent performance.*

5.1.3 Experiment Three

The purpose of this experiment was to test hypothesis 1 by applying the POD method and the QPOD method to a data set pertaining to a three-dimensional fluid system. As outlined in the methodology, this experiment applied the POD method and the QPOD method to six data sets obtained using stereo particle image velocimetry (SPIV) measurements for the flow over a forward facing step. The measurements were taken at six different stations downstream of the forward facing step.

Figure 5.6 shows the proposed flow, as suggested by Eppink [112]. According to Ep-

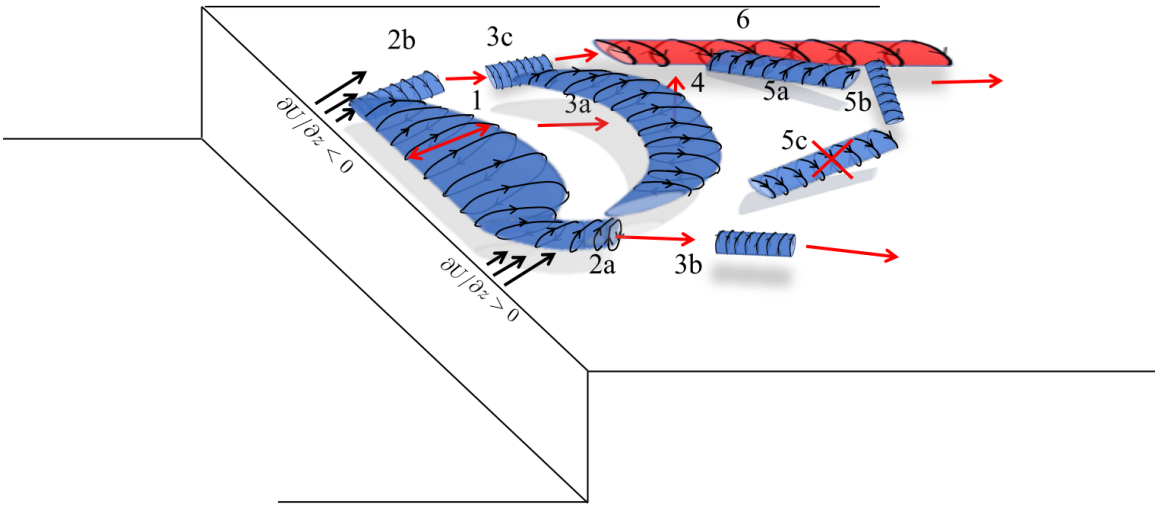


Figure 5.6: Proposed flow over a forward facing step as suggested by Eppink [112].

pink [112], the forward facing step causes the flow to separate resulting in two-dimensional separation bubbles. At the same time, the stationary cross flow introduces a modulation to the two-dimensional separation bubbles causing them to become three-dimensional. Consequently, the flow becomes three-dimensional which results in the formation of vortices, some of which exhibit complex vortex shedding. Figure 5.6 shows the primary vortex, vortex 1, which begins in the span-wise direction and evolves downstream into vortex 3a. Even further downstream vortex 3a turns into vortex 5a and 5b which are are oriented stream-wise However, figure 5.6 also shows vortex 6, which is a stationary vortex that interacts with the shedded vortex, vortex 3c.

The first result of this experiment is a qualitative comparison of the singular values resulting from applying the POD and QPOD methods at each station. Figure 5.7 shows the POD singular values, $\{\sigma_i(u^{\mathrm{d}})\}_i$, and the QPOD singular values, $\{\sigma_i^{\mathrm{q}}(u^{\mathrm{dq}})\}_i$, at each station. Looking at the plots in figure 5.7, it was observed that the QPOD singular values descend faster than the POD singular values.

Observation 28. *The QPOD singular values descend more rapidly than their POD counterpart. This is observed at all six stations.*

The resulting singular values at each station were then used to compute the normalized distance metric. Figure 5.8 shows the computed normalized distance metric at each station. The normalized distance metric scores were found to be rather low since η_{n} was in the order of 10^{-4} at all stations. Figure 5.8 shows that the normalized distance metric tends to increase downstream, but not by a significant amount.

Observation 29. *As the flow evolved downstream, the normalized distance metric η_n increased. An increase in η_n indicates a greater divergence between the POD results and QPOD results. It is also interesting to note that the flow becomes more turbulent as it evolves downstream.*

Before the performance metrics were computed, a detailed inspection of the results

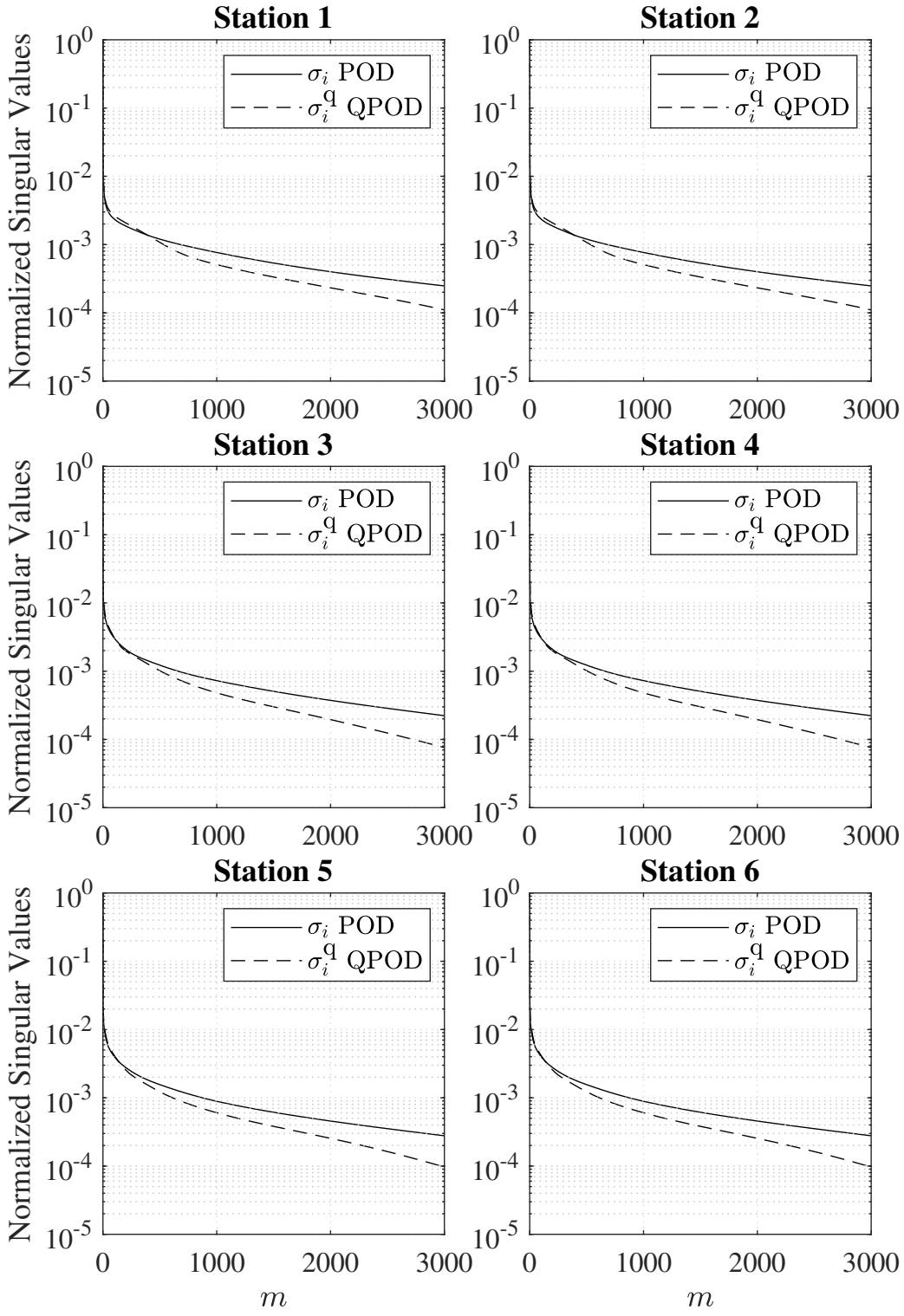


Figure 5.7: Normalized singular values of the QPOD method and the normalized singular values of the POD method plotted at each station downstream of the forward facing step.

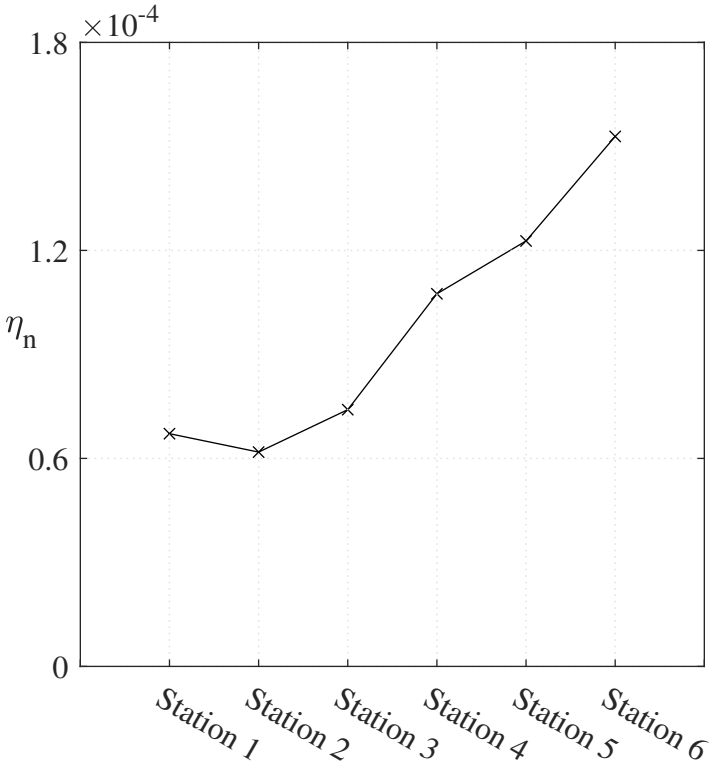


Figure 5.8: Normalized distance metric computed for the six different data sets. Each data set is collected at a different location downstream of the flow. The normalized distance metric between the POD method and the QPOD method is computed for each data set. The data sets are labeled according to their station number which increases further downstream of the forward facing step.

revealed that the mean of the flow contributes to 95% of the flow. Therefore, without any loss to the original aim of this investigation, the performance metrics were slightly modified such that more emphasis is given to the ability of the methods to approximate the fluctuations of the flow. The fluctuation errors are denoted by $\tilde{\epsilon}_m^{\mathrm{r}}$, $\tilde{\epsilon}_m^{\mathrm{q}}$, and $\tilde{\epsilon}_m$ and were used instead of the previously established errors ϵ_m^{r}, ϵ_m^{q}, and ϵ_m. In this manner, a better contrast between the POD and the QPOD is established. By comparing the ability of each method to reconstruct deviations of the flow from the mean, i.e. comparing the fluctuation errors, the magnitude of th mean and its effects are muted from the analysis. The POD fluctuation error is defined as

$$\tilde{\epsilon}_m^{\mathrm{r}} = \frac{\|u^{\mathrm{d}} - \bar{u}^{\mathrm{d}} - \sum_{i=2}^{m} \phi_i(X) a_i\|}{\|u^{\mathrm{d}} - \bar{u}^{\mathrm{d}}\|} \times 100, \tag{5.8}$$

where $\bar{u}^{\mathrm{d}} = \mathbb{E}(u^{\mathrm{d}})$ is the average over all snapshots. Since the data is not centered, the first mode is the normalized average ($\phi_1 = \bar{u}^{\mathrm{d}}/\|\bar{u}^{\mathrm{d}}\|_{\mathrm{F}}$), therefore the POD fluctuation error can be restated as

$$\begin{aligned} \tilde{\epsilon}_m^{\mathrm{r}} &= \frac{\|u^{\mathrm{d}} - \sum_{i=1}^{m} \phi_i(X) a_i\|}{\|u^{\mathrm{d}} - \bar{u}^{\mathrm{d}}\|} \times 100 \\ &= \frac{\|u^{\mathrm{d}} - u_m^{\mathrm{d}}\|}{\|u^{\mathrm{d}} - \bar{u}^{\mathrm{d}}\|} \times 100. \end{aligned} \tag{5.9}$$

By following the same steps, an expression for the QPOD fluctuation error is given by

$$\begin{aligned} \tilde{\epsilon}_m^{\mathrm{q}} &= \frac{\|u^{\mathrm{dq}} - \sum_{i=1}^{m} \phi_i^{\mathrm{q}}(X) a_i^{\mathrm{q}}\|}{\|u^{\mathrm{dq}} - \bar{u}^{\mathrm{dq}}\|} \times 100 \\ &= \frac{\|u^{\mathrm{dq}} - u_m^{\mathrm{dq}}\|}{\|u^{\mathrm{dq}} - \bar{u}^{\mathrm{dq}}\|} \times 100. \end{aligned} \tag{5.10}$$

Figure 5.9, shows the fluctuation errors associated with the POD and QPOD methods plotted against the number of modes used to create the approximations. It is clear from figure 5.9 that the QPOD method was superior to the POD method at every station.

Observation 30. *Without exception, at every station, the QPOD method produced better approximations of the flow fluctuations than the POD method.*

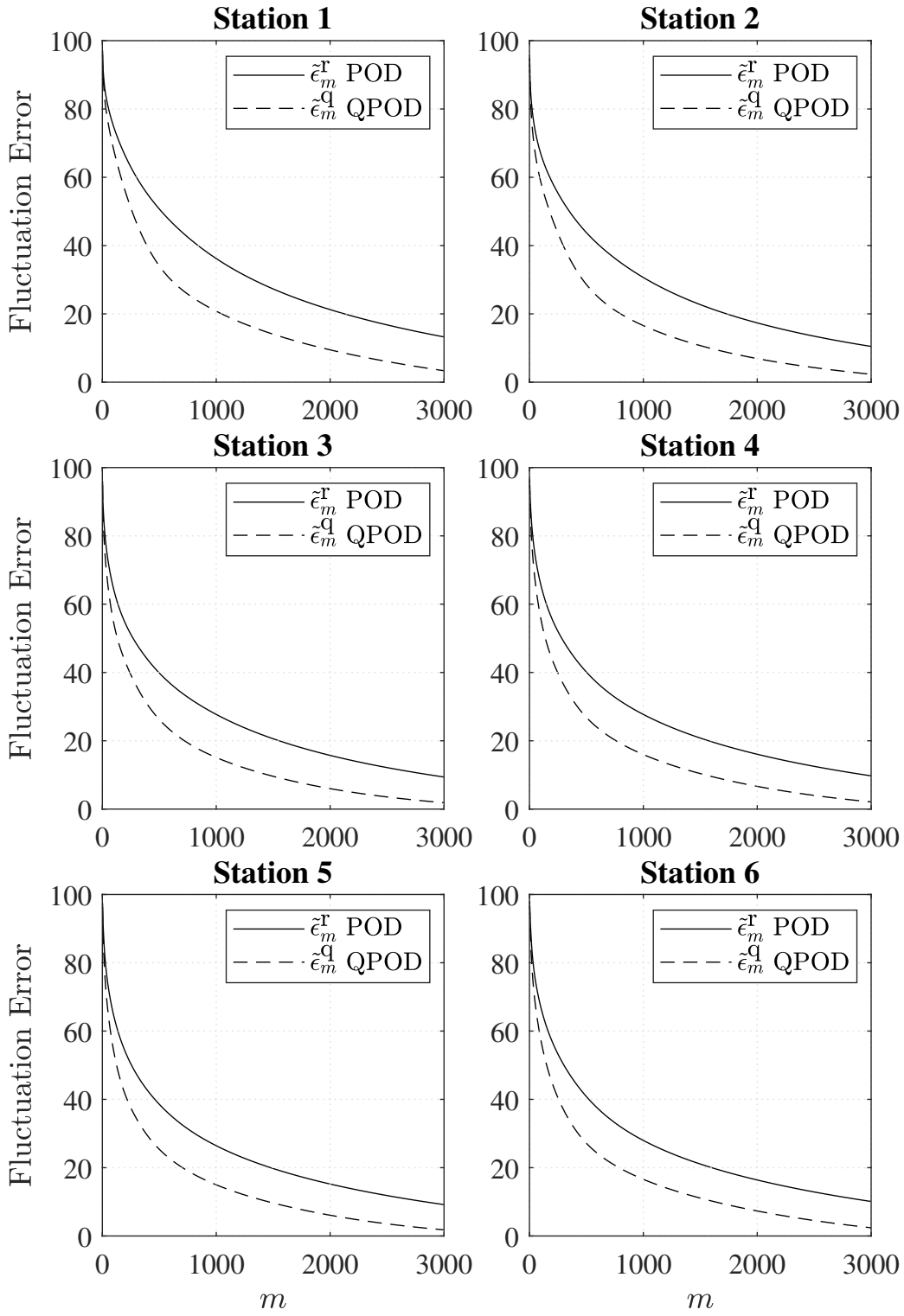


Figure 5.9: The fluctuation error associated with the POD method, $\tilde{\epsilon}_m^{\mathrm{r}}$, and the fluctuation error associated with QPOD method, $\tilde{\epsilon}_m^{\mathrm{q}}$, for the flow over a forward facing step.

Observation 31. *On average, the QPOD method resulted in 20% fluctuation error using approximately one thousand less modes than the POD method. This can be seen by drawing a horizontal line at the 20% mark in the plots shown in figure 5.9.*

Mode visualization

Visualizing the POD and QPOD modes helps in identifying patterns that might help explain the evolution of the flow over time. In fact, the POD method was derived by Lumley [1] for the purpose of extracting flow patterns. In this section, the POD and QPOD modes were visualized and compared. This was done in order to compare the capabilities of the POD and QPOD methods to extract meaningful patterns from a data set pertaining to a fluid system.

Since the POD and QPOD methods were applied at every station, this resulted in a large set of POD modes (also known as coherent structures), and a large set of QPOD modes (defined in this thesis as the quaternion coherent structures). Due to the large number of modes available, only a select few were presented here, however, the remaining can be found at `https://github.com/yissac/kaleidoscope.git`.

As previously noted in observation 12, every QPOD mode ϕ_i^{q} consists of four components $\phi_i^{\mathrm{rq}}, \phi_i^{\mathrm{iq}}, \phi_i^{\mathrm{jq}}$, and ϕ_i^{kq}. On the other hand, every POD mode ϕ_i consists of three components $\phi_i^{\mathrm{u}}, \phi_i^{\mathrm{v}}$, and ϕ_i^{w}. Table 3.1 summarizes how the components of the POD modes and QPOD modes are extracted. The summary is repeated here as table 5.2.

POD	**QPOD**
$u^{\mathrm{d}}{u^{\mathrm{d}}}^{\top}\phi_i(X) = \phi_i(X)\lambda_i$	$u^{\mathrm{dq}}{u^{\mathrm{dq}}}^{\star}\phi_i^{\mathrm{q}}(X) = \phi_i^{\mathrm{q}}(X)\lambda_i^{\mathrm{q}}$
$\{\lambda_i = (\sigma_i)^2\}_{i=1}^m$	$\{\lambda_i^{\mathrm{q}} = (\sigma_i^{\mathrm{q}})^2\}_{i=1}^m$
$\phi_i^u = \mathrm{row}_{j=1}^{n}(\phi_i)$	$\phi_i^{\mathrm{rq}} = \mathrm{real}(\phi_i^{\mathrm{q}})$
$\phi_i^v = \mathrm{row}_{j=n+1}^{n\times 2}(\phi_i)$	$\phi_i^{\mathrm{iq}} = \mathrm{imag}_{\boldsymbol{i}}(\phi_i^{\mathrm{q}})$
$\phi_i^w = \mathrm{row}_{j=n\times 2+1}^{n\times 3}(\phi_i)$	$\phi_i^{\mathrm{jq}} = \mathrm{imag}_{\boldsymbol{j}}(\phi_i^{\mathrm{q}})$
	$\phi_i^{\mathrm{kq}} = \mathrm{imag}_{\boldsymbol{k}}(\phi_i^{\mathrm{q}})$

Table 5.2: A summary highlighting how the components of the POD and QPOD modes are extracted.

Figure 5.10 shows the components of the first POD mode (not counting the average mode) and the components of the first QPOD mode (not counting the average mode) at station one.

Theorem 2 states that when quaternion eigenvectors are multiplied by a quaternion scalar, the result is also an eigenvector. Therefore, associated with every eigenvalue λ_i^{q} is a set of eigenvectors. The set of eigenvectors was termed the kaleidoscope set, where all the vectors in a given kaleidoscope set relate to one another via multiplication with a quaternion scalar, which is equivalent to rotating every entry of a given eigenvector in a four dimensional space. This was briefly alluded to in section 4.5.4. Figure 5.11 shows two eigenvectors in the same kaleidoscope set associated with the first eigenvalue, ϕ_1^{q} and $\phi_1^{\mathrm{q,rot}}$, such that

$$\phi_1^{\mathrm{q,rot}} = \phi_1^{\mathrm{q}} w, \tag{5.11}$$

where $w = -0.42 + 0.39\boldsymbol{i} - 0.68\boldsymbol{j} + 0.45\boldsymbol{k}$. Both ϕ_1^{q} and $\phi_1^{\mathrm{q,rot}}$ belong to the kaleidoscope set $\kappa(\lambda_1)$.

In fact, there are infinitely many equivalent vectors, all of which relating to each other, and all belonging to the same kaleidoscope set $\kappa(\lambda_i^{\mathrm{q}})$. Therefore, any vector in $\kappa(\lambda_i^{\mathrm{q}})$ can be considered as the first QPOD mode. `https://github.com/yissac/kaleidoscope`.

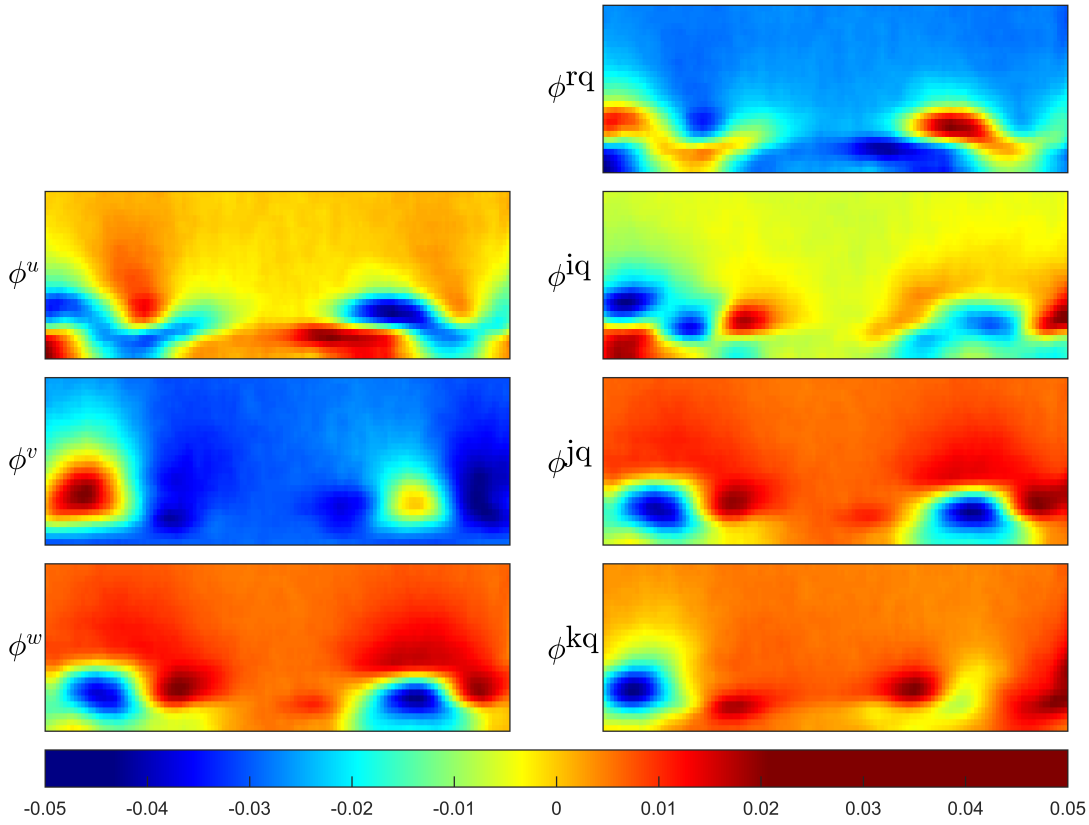


Figure 5.10: The first column to left shows the components of the first POD mode ϕ_1. The column to the right shows the components of the first QPOD mode ϕ_1^{q}. The modes were extracted by applying the POD and QPOD methods to the PIV data at station two.

`git` can be used to visualize the QPOD modes when a continuous unit quaternion multiplication is applied to them, creating a kaleidoscope effect.

Observation 32. *Every QPOD mode belongs to an infinite set of modes termed the Kaleidoscope set. When the components of a QPOD mode are visualized as they are being multiplied by a continuous function $w(t)$ as follows*

$$\phi_i^{q,rot} = \phi_i^q w(t), \tag{5.12}$$

where $w(t) \in \mathbb{H}$ and $\|w(t)\| = 1$, a kaleidoscope effect happens.

The fact that every QPOD mode contains four components that can be rotated adds

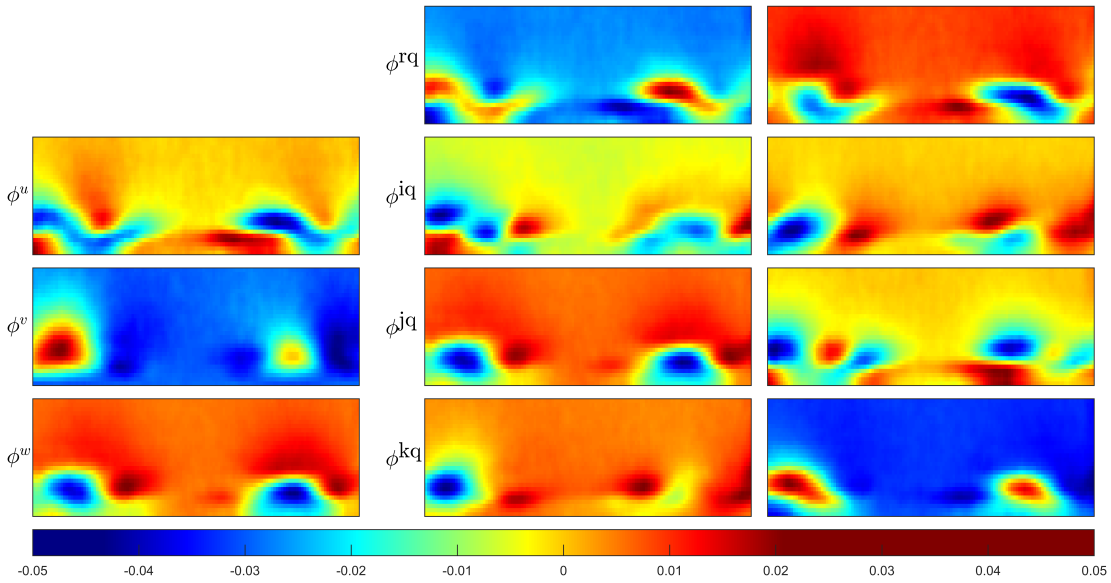


Figure 5.11: The first column to left shows the components of the first POD mode ϕ_1. The middle column shows the components of the first QPOD mode ϕ_1^{q}. The column on the right shows the components of an equivalent rotated version of first QPOD mode $\phi_1^{\mathrm{q,rot}} = \phi_1^{\mathrm{q}} w$. The modes were extracted by applying the POD and the QPOD methods to the PIV data at station two.

extra complexity. This makes it difficult to:

1. Interpret the meaning of the QPOD modes, its associated components, and the possible rotations associated with them.

2. Compare the information captured between the POD modes and QPOD modes.

Hence, the magnitude of the components associated with the i^{th} mode are defined as follows

$$M_{i,j} = \left(\left(\phi^{u}_{i,j}\right)^2 + \left(\phi^{u}_{i,j}\right)^2 + \left(\phi^{u}_{i,j}\right)^2 \right)^{\frac{1}{2}}, \tag{5.13}$$

$$M^{\mathrm{q}}_{i,j} = \left(\left(\phi^{\mathrm{rq}}_{i,j}\right)^2 + \left(\phi^{\mathrm{uq}}_{i,j}\right)^2 + \left(\phi^{\mathrm{vq}}_{i,j}\right)^2 + \left(\phi^{\mathrm{wq}}_{i,j}\right)^2 \right)^{\frac{1}{2}}. \tag{5.14}$$

where $M^{\mathrm{r}}_i, M^{\mathrm{q}}_i \in \mathbb{R}^n$ and j resembles the j^{th} entry associated with the vectors m_i and m^{q}_i. The definition of M^{r}_i and M^{q}_i enables a direct comparison between the POD and QPOD

modes. This is accomplished by taking the difference between M_i^{r} and M_i^{q} as follows

$$M_i = M_i^{\mathrm{r}} - M_i^{\mathrm{q}}. \tag{5.15}$$

Figure 5.12 shows a comparison between the first POD mode and the first QPOD mode at the second station, the QPOD mode is considerably sharper and less blurry.

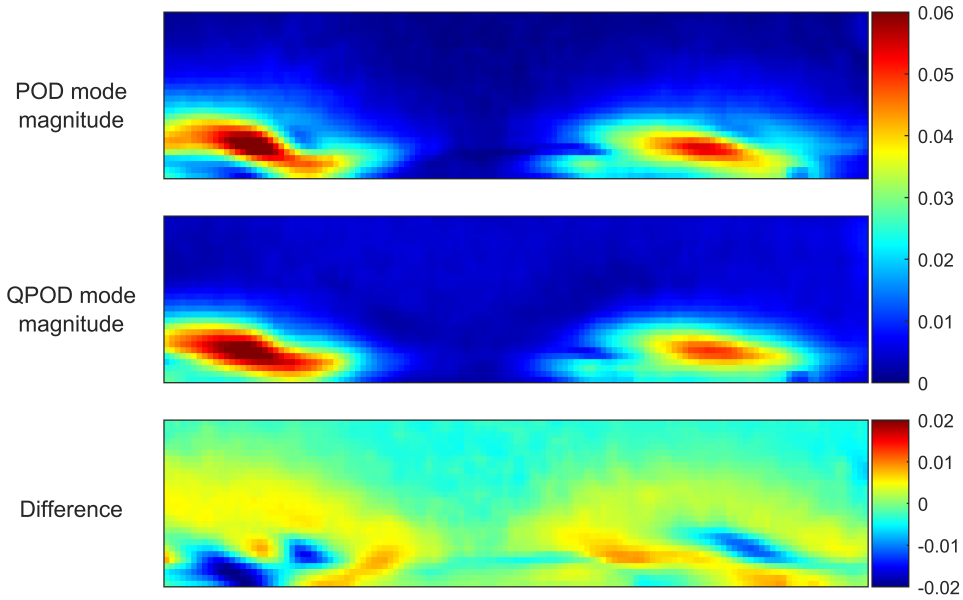


Figure 5.12: The magnitude of the first POD mode and the magnitude of the first QPOD mode are displayed in the first two rows. The third row displays their difference. Results shown are for station one.

Observation 33. *The flow features extracted via the QPOD method are sharper and with less blur around the edges as compared to the flow features extracted via the POD method.*

Figure 5.13 compares the second POD mode and the second QPOD mode obtained at the second station. Figure 5.13 shows a greater discrepancy between the patterns extracted using each method as compared to figure 5.12. The first and second modes extracted using the POD method are very similar.

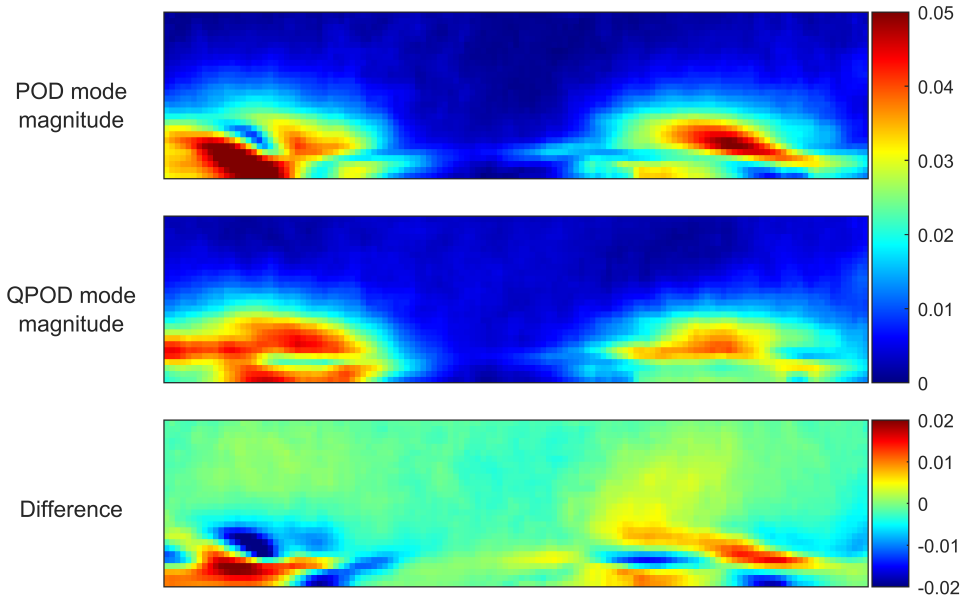


Figure 5.13: The magnitude of the second POD mode and the magnitude of the second QPOD mode are displayed in the first two rows. The third row displays their difference. Results shown are for station one.

5.1.4 Experiment Four

In chapter 4, it was mentioned that a CFD grid was generated and the flow around a three-dimensional cylinder was numerically computed at different Reynolds numbers. The POD and QPOD methods were then applied to the outputs of every simulation ran. This experiment utilized the snapshot implementation of the POD method and the quaternion snapshot implementation of the QPOD method. Hence, this experiment tests the following hypotheses:

1. Hypothesis 1: compare the POD and the QPOD methods as they apply to three dimensional fluid systems.

2. Hypothesis 2: verify the scalability of the QPOD method.

As with all the previous experiments, it is important to look for four results:

- Normalized distance metric, η_n, for each simulation ran.

- Singular values, $\{\sigma_i(u^{\mathrm{d}})\}_i$ and $\{\sigma_i^q(u^{\mathrm{dq}})\}_i$.
- Performance metrics, ϵ_m^{r} and ϵ_m^{q}.
- Mode comparison and visualization.

Figure 5.14 shows the normalized distance metric computed and plotted for each simulation case. The plot shows a very close agreement between the POD and QPOD results

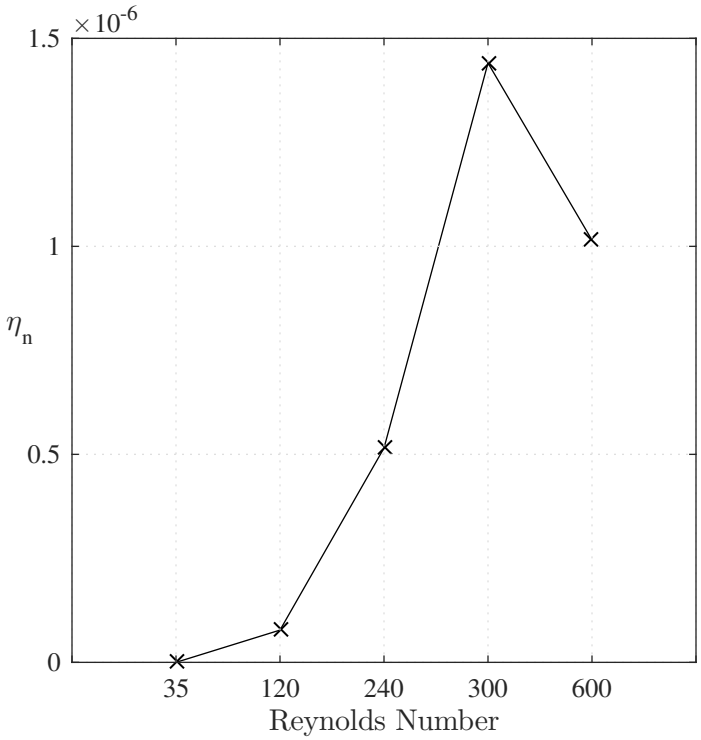


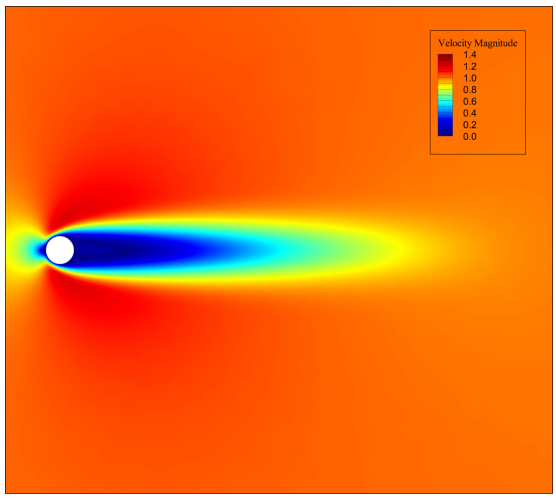
Figure 5.14: Normalized distance metric η_{n} computed for each simulation ran. The Reynolds number was increased from one simulation to the next.

for all cases as the normalized distance metric η_n was found to be in the order of 10^{-6}. However, for the case where the Reynolds number was set to 35, both the POD and the QPOD methods gave identical results and $\eta_n = 0$.

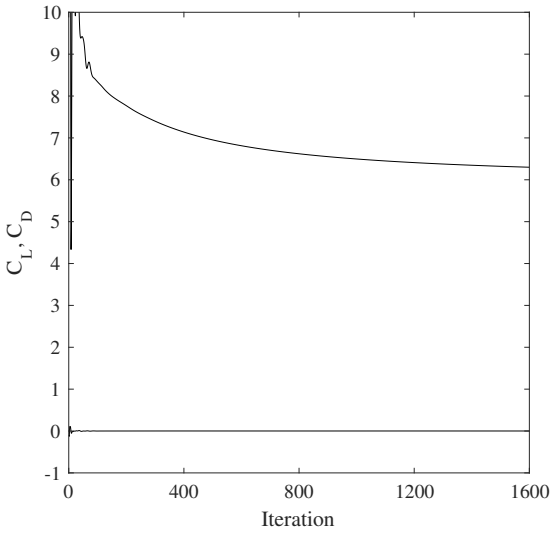
Re = 35

In accordance with the previously mentioned literature, at Reynolds number below the critical Reynolds number, the flow developed into a steady flow. Figure 5.15b shows the

lift and drag coefficients converging to their steady state values as the flow becomes steady. Figure 5.15a shows the contour plots of the velocity magnitude of the developed flow at iteration 1550.



(a) Simulation snapshot of the flow at iteration 1550. Contour plot of the velocity magnitude $\|U\|$ is shown, where $\|U\| = \sqrt{u^2 + v^2 + w^2}$. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.



(b) Lift and drag coefficients on the cylinder versus iteration number at Reynolds number 35.

Figure 5.15: Reynolds number 35

After the flow was allowed to develop, and the solution correctly converged to the steady state solution, the POD and QPOD methods were applied to 50 consecutive snapshots of the developed flow. In this case, both the POD and the QPOD gave identical results, as can be seen in figure 5.16. Not only are the modes identical, but they also resulted in the same approximations. Moreover, the QPOD modes were found to be identical to the POD modes,

Re = 120

As the Reynolds number was increased beyond the critical Reynolds number, a Hopf bifurcation occurred in which the stable, steady state solution became unstable, and a two dimensional, periodic attractor was born. This specific attractor for the two or three dimensional cylinder at this range of Reynolds number is known as the Benard-Von Karmen

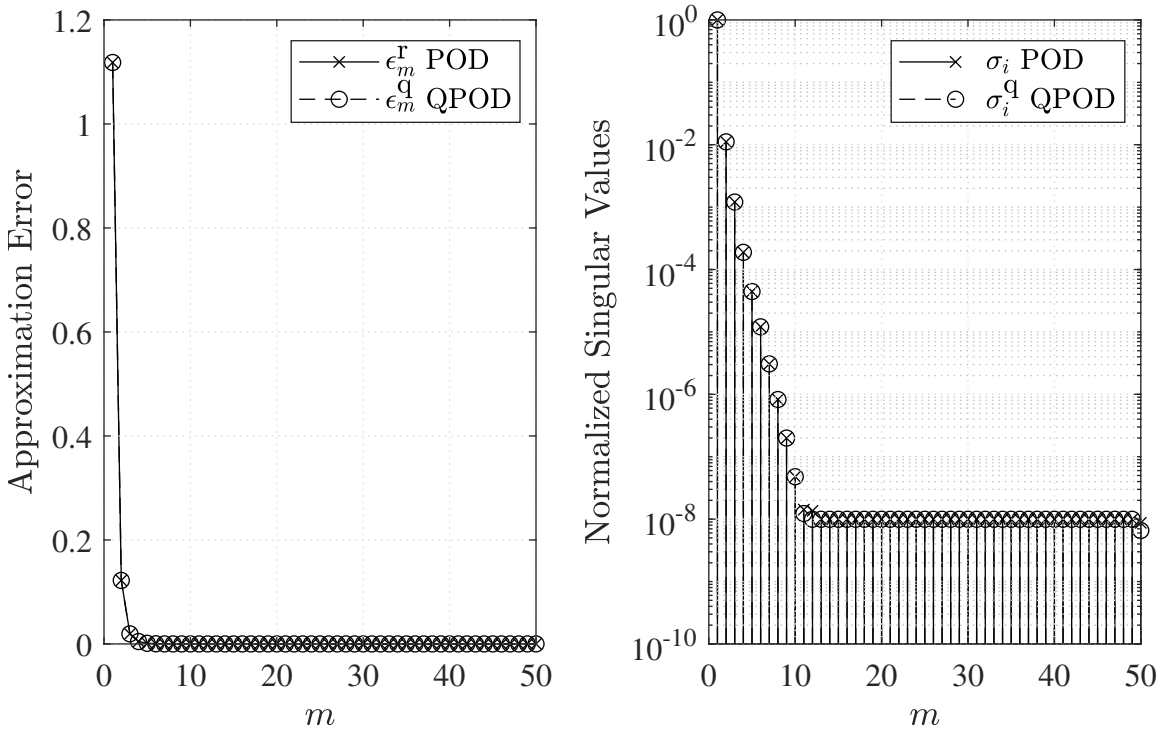
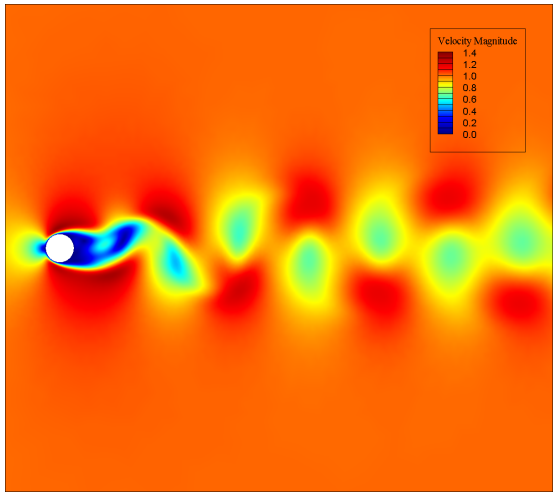
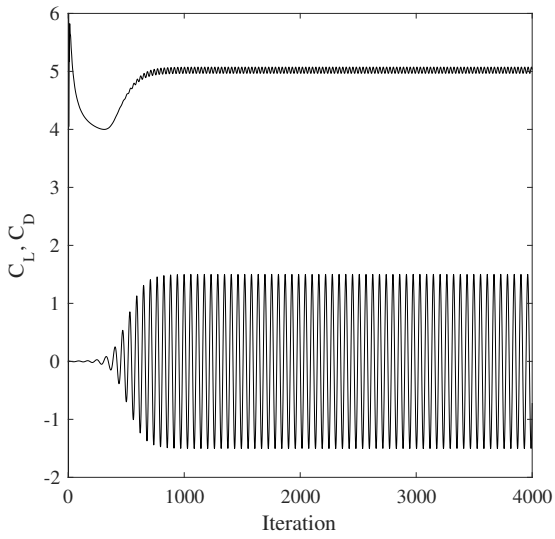


Figure 5.16: The plot on the left shows the errors of the POD reduced order models and the QPOD reduced order models versus the number of modes used to generate them. The plot on the right shows the normalized singular values of each method.

vortex street. The result of simulating the three dimensional cylinder at Reynolds number 120 can be seen in figure 5.17. A snapshot of the simulation at iteration 6200 showing the Benard-Von Karmen vortex street is shown in figure 5.17a. The contour plot of the velocity magnitude shows the same structure that has been shown in the previous literature. In



(a) Simulation snapshot of the flow at iteration 6200. Contour plot of the velocity magnitude $\|U\|$ is shown, where $\|U\| = \sqrt{u^2 + v^2 + w^2}$. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.



(b) Lift and drag coefficients versus iteration number.

Figure 5.17: Simulation results at Reynolds number 120.

figure 5.17b, the lift and drag coefficients are plotted against the iteration number. It can be seen that when enough time is allowed for the flow to develop, the flow becomes periodic as expected.

The POD and the QPOD methods were then applied to a subset of the data. Specifically, 120 snapshots were chosen in the periodic region and the flow field variables of every 10^{th} iteration between iteration number 5010 to 6200 was used. Figure 5.18 shows the results of applying the POD and the QPOD methods. Both methods produce almost identical results. The singular values are almost identical and the approximation errors of both methods are equal.

Figure 5.19 shows the average mode and the first three modes. Since both methods produce almost identical modes only one subset of the modes is shown here. The modes

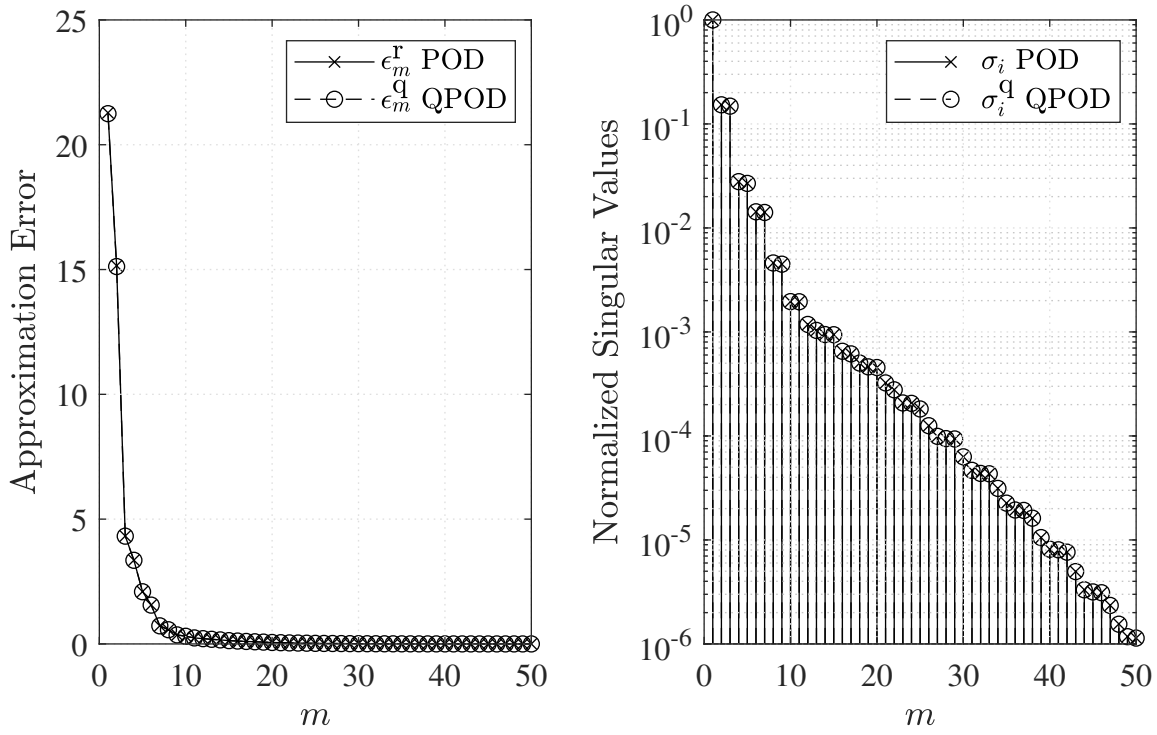
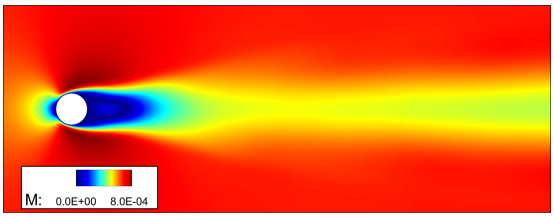
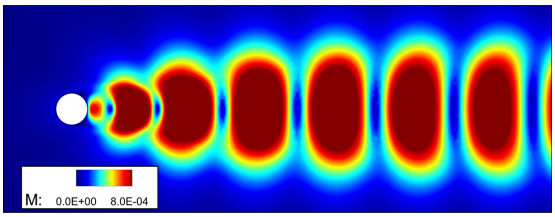


Figure 5.18: The plot on the left shows the errors of the POD reduced order models and the QPOD reduced order models versus the number of modes used to generate them. The plot on the right shows the normalized singular values of each method.

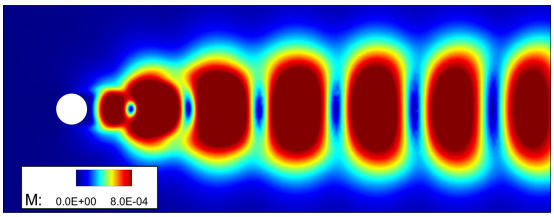
exhibit translational symmetry in the Y-direction since the flow is strictly two dimensional, i.e. $v = 0$, albeit machine error. Hence, only the contours on a planar slice along the Y-axis at Y=6 are shown. The modes shown are to be in accordance with the ones shown by



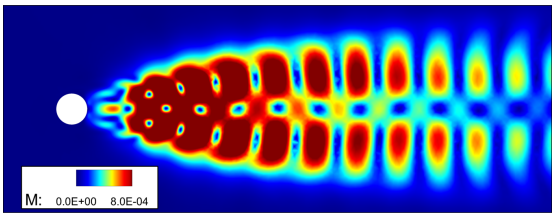
(a) Contours of the velocity magnitude of the average flow. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.



(b) Contours of the velocity magnitude of the first mode, M_1^{r}. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.



(c) Contours of the velocity magnitude of the second mode, M_2^{r}. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.



(d) Contours of the velocity magnitude of the third mode, M_3^{r}. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.

Figure 5.19: Subset of the POD modes generated using the data set with $Re = 120$. The figures show contours of $M_i^{\mathrm{r}} = \sqrt{(\phi_i^u)^2 + (\phi_i^v)^2 + (\phi_i^w)^2}$.

Deanne in 1991 [125]. By comparing to the notes of Noack et al. [127], it was also verified that the modes with $i = 1, 2, 5, 6, \ldots$ are anti-symmetric with the x-axis,

$$\phi_i^u(x, -z) = -\phi_i^u(x, z),$$

$$\phi_i^w(x, -z) = +\phi_i^w(x, z),$$

and the modes with $i = 3, 4, 7, 8, \ldots$ are symmetric with the x-axis,

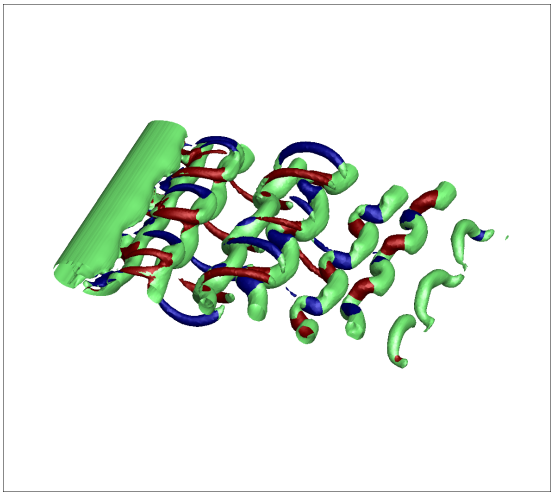
$$\phi_i^u(x, -z) = +\phi_i^u(x, z),$$
$$\phi_i^w(x, -z) = -\phi_i^w(x, z).$$

The reader should note that figure 5.19 does not show ϕ_i^u, ϕ_i^v, or ϕ_i^w, instead the contours of the magnitude of the modes are shown.

Re = 240

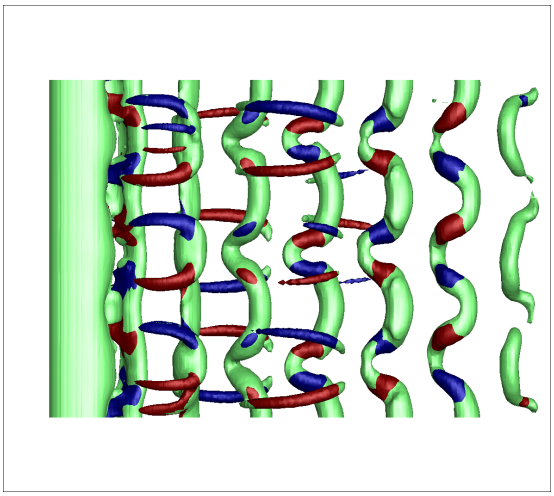
As the Reynolds number is further increased, bifurcations keep occurring in accordance to the universal scaling laws of nonlinear systems [119, 120]. In the case of the three dimensional cylinder, the two dimensional periodic orbit became unstable and a new three dimensional attractor was born. This is typically observed with the presence of stream-wise vortices and span-wise flow, as discussed in more details by Roshko [116] and Williamson [117, 114]. The result of simulation at Reynolds number 240 can be seen in figure 5.20.

A snapshot of the simulation at iteration 65370 showing the three dimensional flow is shown in figure 5.20. It is common practice to plot the iso-surfaces of the Q-criterion in three-dimensional flows, where vortex structures might exist. Figure 5.20 demonstrates the three-dimensional structures and the three-dimensional flow by plotting the iso-surfaces of the Q-criterion at 0.25. The stream-wise vortices are colored based on the rotational direction of the vortices with red representing clockwise rotation and blue representing counter clockwise rotation. Typically, a modulation to the grid is given in the span-wise direction to excite flow in the span-wise direction allowing for three-dimensional effects to develop. In this simulation, no excitation of such was done, instead the round-off and truncation errors uniformly distributed over the grid served as the excitation required for the flow to become three-dimensional. For this reason, the flow required a large number of iterations for the three-dimensional effects to develop. This can be seen in the lift and drag

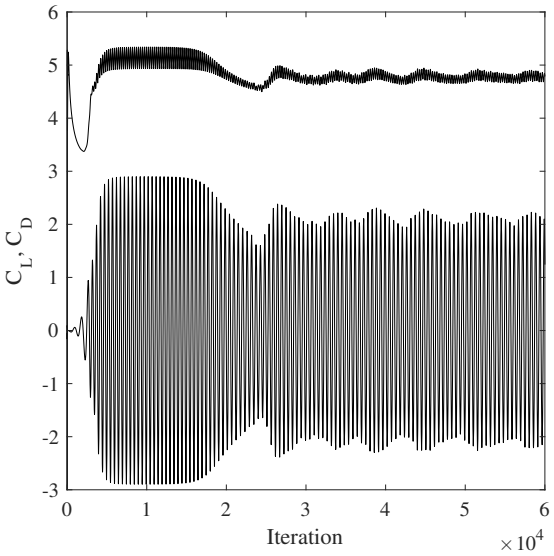


(a) View showing the iso-surface of the Q-criterion set to 0.25 for snapshot 65370 at Reynolds number 240.

(b) Side view showing the iso-surface of the Q-criterion set to 0.25 for snapshot 65370 at Reynolds number 240.



(c) Top view showing the iso-surface of the Q-criterion set to 0.25 for snapshot 65370 at Reynolds number 240.



(d) Lift and drag coefficient versus iteration number at Reynolds number 240.

Figure 5.20: Simulation results for $Re = 240$.

coefficients plot shown in figure 5.20d. As the solution was marched forward in time, the three dimensional instabilities grew, the stable periodic oscillations of the Karman street became unstable, and the solution converged towards a three-dimensional periodic orbit. This caused the lift and drag coefficients to deviate from the two-dimensional Karman street pattern at around iteration number 18,000 and beyond. Another characteristic of three-dimensionality is the growth in span-wise velocity. At Reynolds numbers where the flow is strictly two-dimensional, as is the case at $Re = 120$, the span-wise velocity, v, always remained very close to 0. However, at $Re = 240$, as the three dimensional instabilities developed, the magnitude of the span-wise velocity became substantial.

The POD and the QPOD snapshot methods were applied to a subset of the data. Specifically, 615 snapshots were chosen. The flow field variables of every 10^{th} iteration between iteration number 59230 and 65380 was used. Figure 5.21 shows the results of applying the POD and the QPOD methods. Both methods produced almost identical results. The singular values are almost identical and both methods produced equal approximations.

Figure 5.22 shows the average mode and the first three POD modes. The contours of the velocity magnitudes of the modes are shown, where the velocity magnitudes for the POD modes are computed as

$$M^{\mathrm{r}}_{i,j}(X) = \sqrt{\left(\phi^{u}_{i,j}\right)^2 + \left(\phi^{v}_{i,j}\right)^2 + \left(\phi^{w}_{i,j}\right)^2}, \tag{5.16}$$

and the velocity magnitude of the QPOD modes are computed as

$$M^{\mathrm{q}}_{i,j}(X) = \sqrt{\left(\phi^{\mathrm{rq}}_{i,j}\right)^2 + \left(\phi^{\mathrm{uq}}_{i,j}\right)^2 + \left(\phi^{\mathrm{vq}}_{i,j}\right)^2 + \left(\phi^{\mathrm{wq}}_{i,j}\right)^2}. \tag{5.17}$$

When the velocity magnitudes are computed for each mode and the contours plotted, both methods produce almost identical modes. Hence only a subset of the POD modes is shown here. It is important to note that since the flow is three dimensional, the modes are not translationally symmetric along the Y-axis. However, in order to visualize the three dimensional

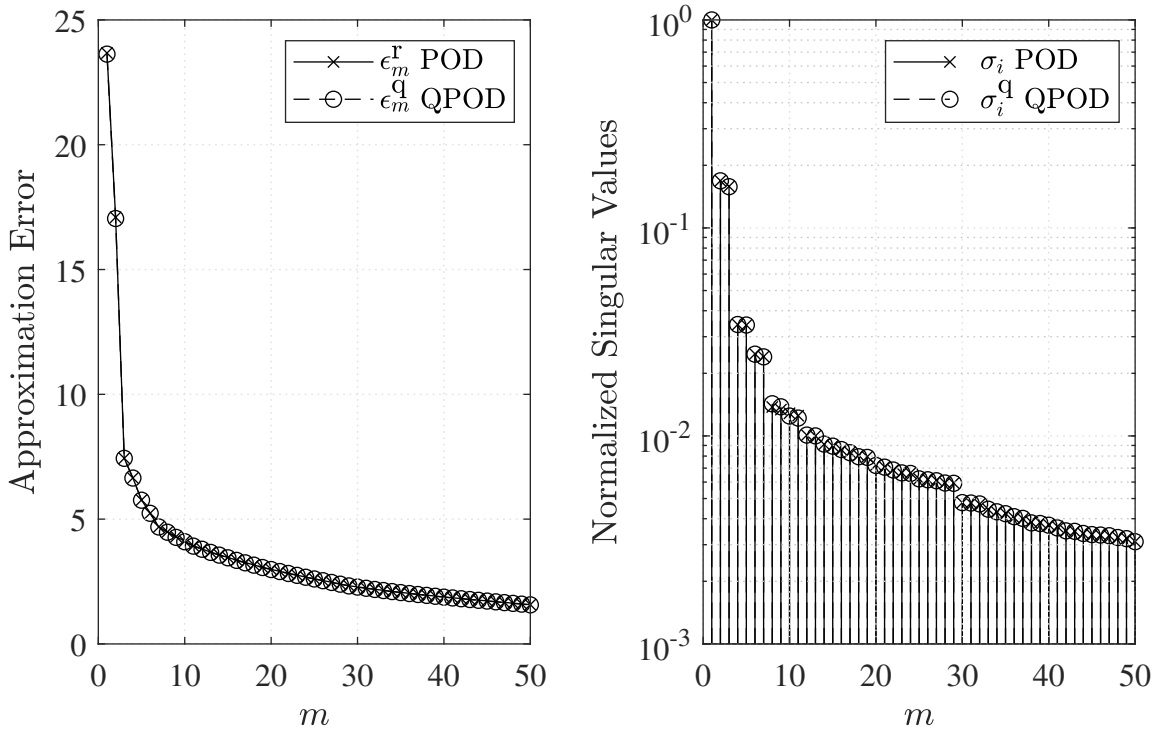
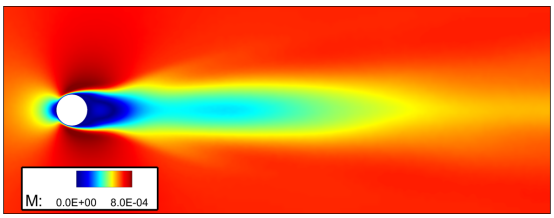
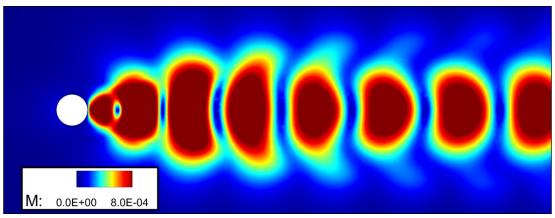


Figure 5.21: Plot to the left shows the ROM errors for the POD and QPOD methods. The plot on the right shows the normalized singular values associated with each mode from each method. Results are for $Re = 240$.

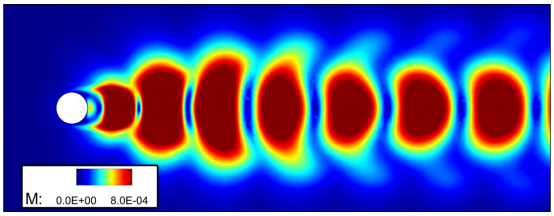
modes, the contours on a slice along the Y-axis at Y=6 was used.



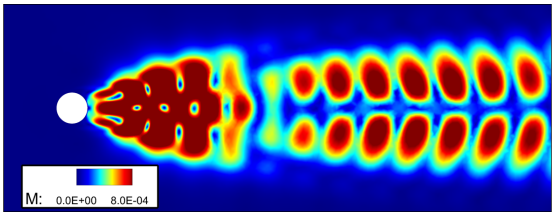
(a) Contours of the velocity magnitude of the average flow. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.



(b) Contours of the velocity magnitude of the first mode, M_1^{r}. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.



(c) Contours of the velocity magnitude of the second mode, M_2^{r}. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.



(d) Contours of the velocity magnitude of the third mode, M_3^{r}. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.

Figure 5.22: Subset of the POD modes generated using the data set with $Re = 240$. The figures show contours of $M_i^{\mathrm{r}} = \sqrt{(\phi_i^u)^2 + (\phi_i^v)^2 + (\phi_i^w)^2}$.

Re = 300

As the Reynolds number was further increased, more bifurcations occurred. Mittal and Balachandar [122] show via numerical simulation of the three dimensional unconfined cylinder that the flow becomes three-dimensional sooner at higher Reynolds numbers, which is in accordance with the results shown in figure 5.24. Indeed figure 5.23 shows that three-dimensional effects began to grow around iteration number 8,000, which is much sooner when compared to the flow simulation at $Re = 240$ where three-dimensional effects began around iteration number 18,000. This can be identified by observing the lift and drag coefficients as the solution starts to deviate from the two-dimensional periodic pattern defined by the Benard-Von Karman vortex street.

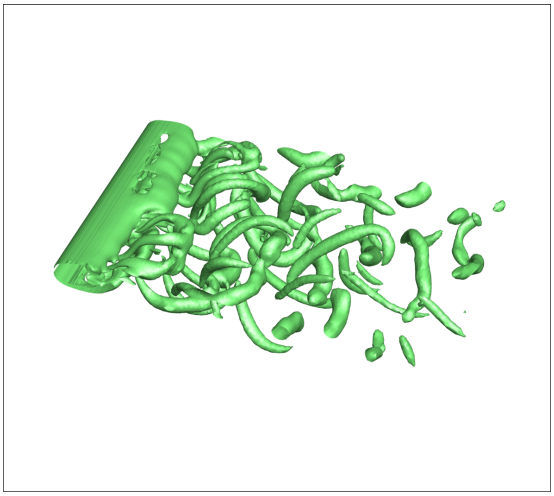
It is common practice to plot the iso-surfaces of the Q-criterion in three-dimensional, flows where vortex structures might exist. Figure 5.20, demonstrates the three-dimensional structures and the three-dimensional flow by plotting the iso-surfaces of the Q-criterion at 0.25. It is interesting to compare the three dimensional structures and flow patterns shown in figure 5.23 for the simulation at $Re = 300$ and the three dimensional structures and flow patterns shown in figure 5.20 for the simulation at $Re = 240$.

The POD and the QPOD snapshot methods were then applied to a subset of the data. Specifically, 1000 snapshots were chosen. The flow field variables of every 10^{th} iteration between iteration number 53,000 and 54,000 was used. Figure 5.24 shows the results of applying the POD and the QPOD methods. Both methods produced almost identical results. The singular values are almost identical and the approximations of both methods almost exactly match.

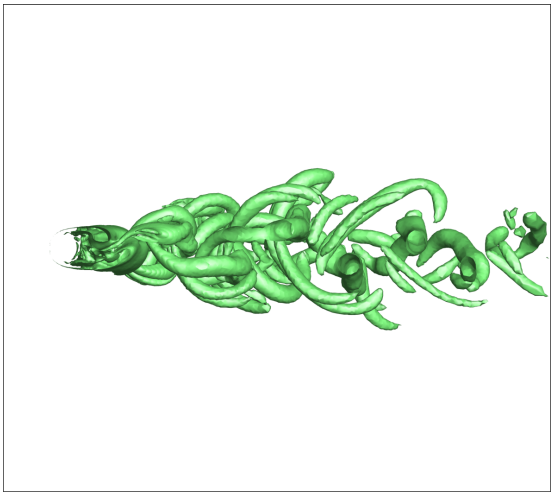
Figure 5.25 shows the average mode and the first three POD modes. The contours of the velocity magnitudes of the modes are shown, where the velocity magnitudes for the POD modes are computed as shown in equation 5.16, and the velocity magnitude of the QPOD modes are computed, as shown in equation 5.17. When the velocity magnitudes are computed for each mode and the contours plotted, both methods produce almost identical modes. Hence, only a subset of the POD modes is shown here. It is important to note that since the flow is three-dimensional, the modes are not translationally symmetric along the Y-axis. However, in order to visualize the three dimensional modes, the contours on a slice along the Y-axis at Y=6 was used.

Re = 600

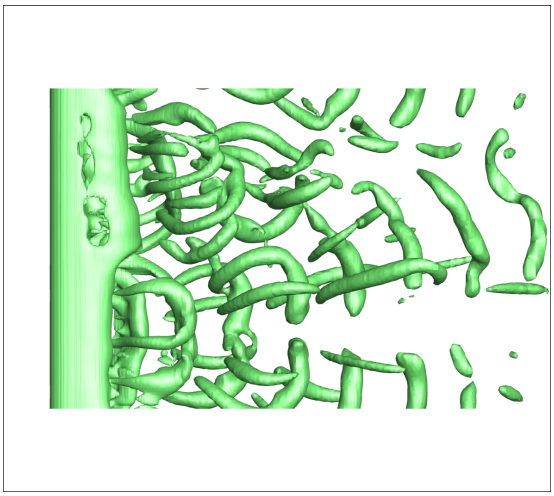
In order to perform a preliminary exploitative study of the potential of the QPOD, another simulation is run at Reynolds number 600. The simulation results are shown in figure 5.26. Figure 5.26d shows the onset of three-dimensionality occurring much sooner, around iteration number 500, when compared to the simulations ran at the lower Reynolds numbers.



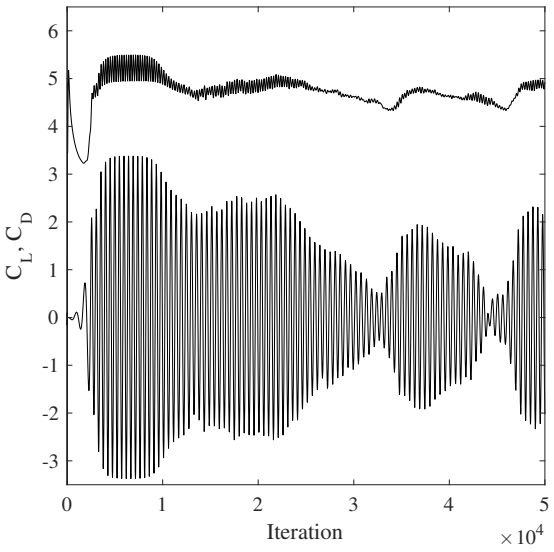
(a) View showing the iso-surface of the Q-criterion set to 0.25 for snapshot 54,000 at Reynolds number 300.



(b) Side view showing the iso-surface of the Q-criterion set to 0.25 for snapshot 54,000 at Reynolds number 300.



(c) Top view showing the iso-surface of the Q-criterion set to 0.25 for snapshot 54,000 at Reynolds number 300.



(d) Lift and drag coefficient versus iteration number at Reynolds number 300.

Figure 5.23: Simulation results for $Re = 300$.

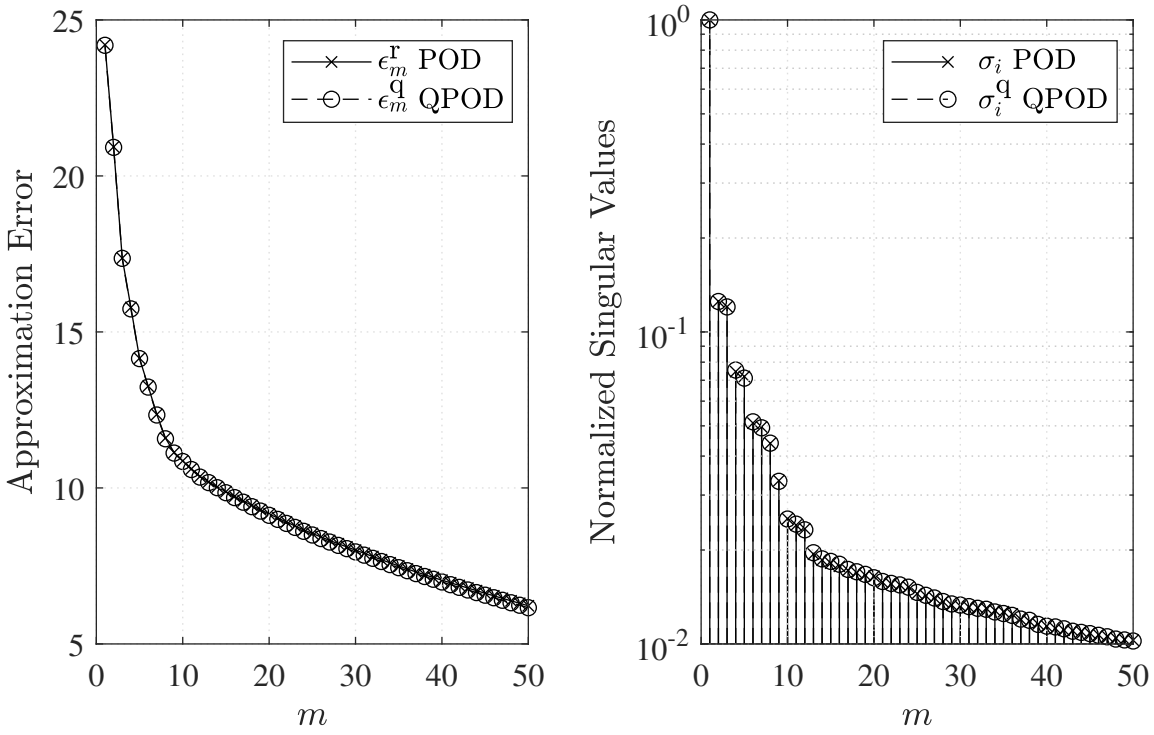
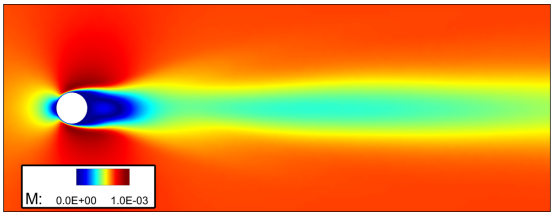
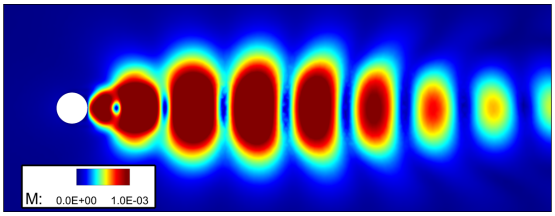


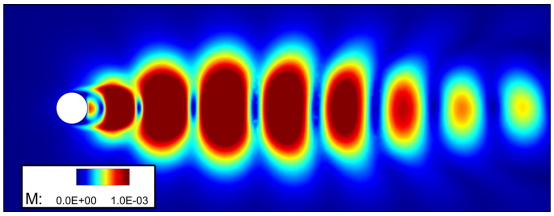
Figure 5.24: Plot to the left shows the ROM errors for the POD and the QPOD methods. The plot on the right shows the normalized singular values associated with each mode from each method. Results are for $Re = 300$.



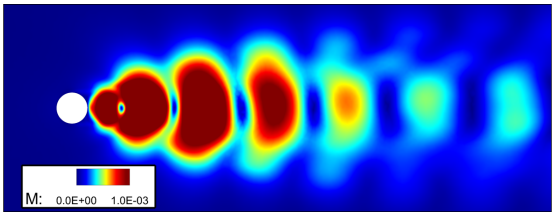
(a) Contours of the velocity magnitude of the average flow. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.



(b) Contours of the velocity magnitude of the first mode, M_1^{r}. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.



(c) Contours of the velocity magnitude of the second mode, M_2^{r}. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.



(d) Contours of the velocity magnitude of the third mode, M_3^{r}. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.

Figure 5.25: Subset of the POD modes generated using the data set with $Re = 300$. The figures show contours of $M_i^{\mathrm{r}} = \sqrt{(\phi_i^u)^2 + (\phi_i^v)^2 + (\phi_i^w)^2}$.

Figure 5.26a shows a snapshot of the simulation flow at iteration number 66,000. The surfaces shown are the iso-surfaces of the Q-criterion at 0.25. Figures 5.26b and figure 5.26c show the side and top view of the same snapshot.

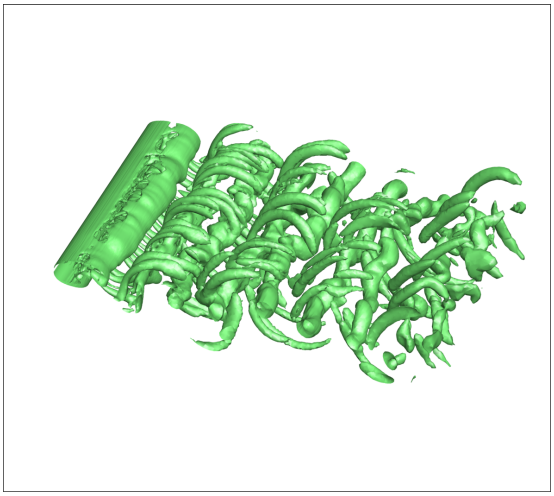
The POD and the QPOD snapshot methods are applied to a subset of the data. Specifically, 1000 snapshots were chosen. The flow field variables of every 10^{th} iteration between iteration number 56,000 and 66,00 was used. Figure 5.27 shows the results of applying the POD and the QPOD methods. Both methods produce almost identical results. The singular values are almost identical and the approximations of both methods are almost equal.

Figure 5.28 shows the average mode and the first three POD modes. The contours of the velocity magnitudes of the modes are shown, where the velocity magnitudes for the POD modes are computed as shown in equation 5.16, and the velocity magnitude of the QPOD modes are computed as shown in equation 5.17. When the velocity magnitudes are computed for each mode and the contours plotted, both methods produce almost identical modes. Hence, only a subset of the POD modes is shown here. It is important to note that since the flow is three dimensional, the modes are not translationally symmetric along the Y-axis. However, in order to visualize the three dimensional modes, the contours on a slice along the Y-axis at Y=6 was used.

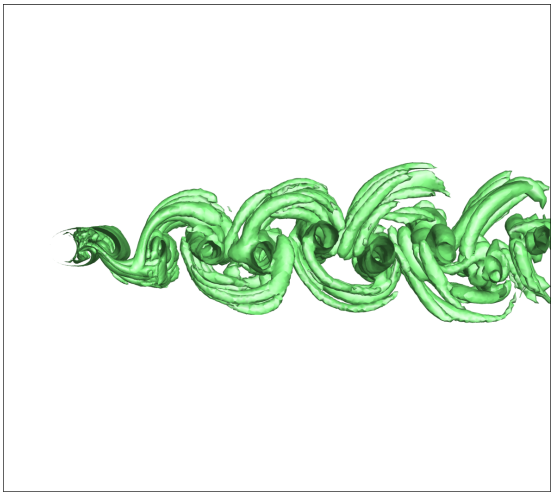
Summary of Results

The grid used in this experiment consisted of 2,271,775 nodes, at every node the three flow field variables were computed and stored. In the simulation run where $Re = 600$, one thousand snapshots were used resulting in a data set comprised of 6,815,325,000 real numbers, around 48 Giga-bytes of storage space. Using the direct implementation on such a data set would not have been possible. However, the quaternion snapshot implementation was able to perform the quaternion proper orthogonal decomposition, and also gave consistent results when compared to the traditional proper orthogonal decomposition.

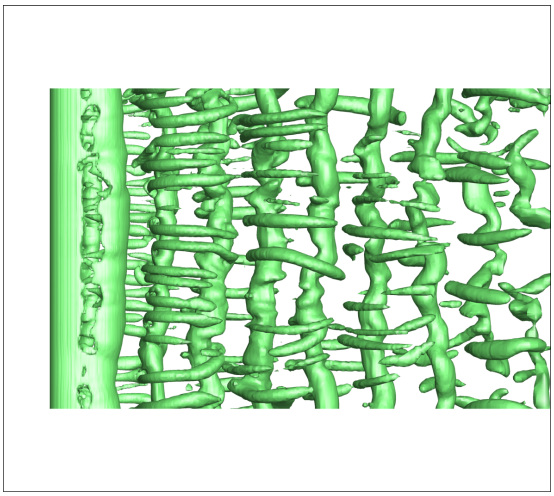
Observation 34. *The quaternion snapshot implementation enables the scalability of the*



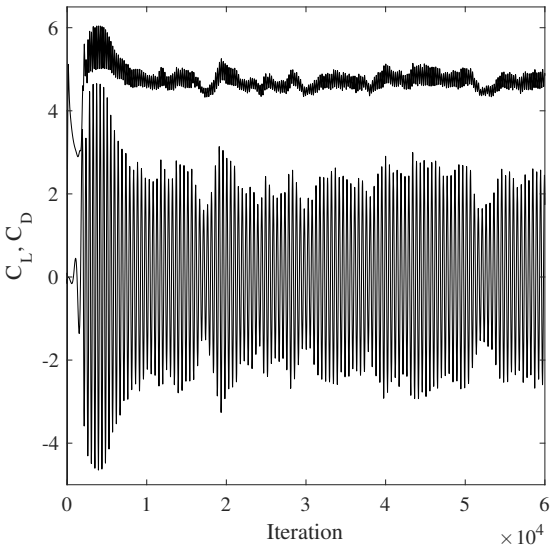
(a) View showing the iso-surface of the Q-criterion set to 0.25 for snapshot 66,000 at Reynolds number 600.



(b) Side view showing the iso-surface of the Q-criterion set to 0.25 for snapshot 66,000 at Reynolds number 600.



(c) Top view showing the iso-surface of the Q-criterion set to 0.25 for snapshot 66,000 at Reynolds number 600.



(d) Lift and drag coefficient versus iteration number at Reynolds number 600.

Figure 5.26: Reynolds number 600

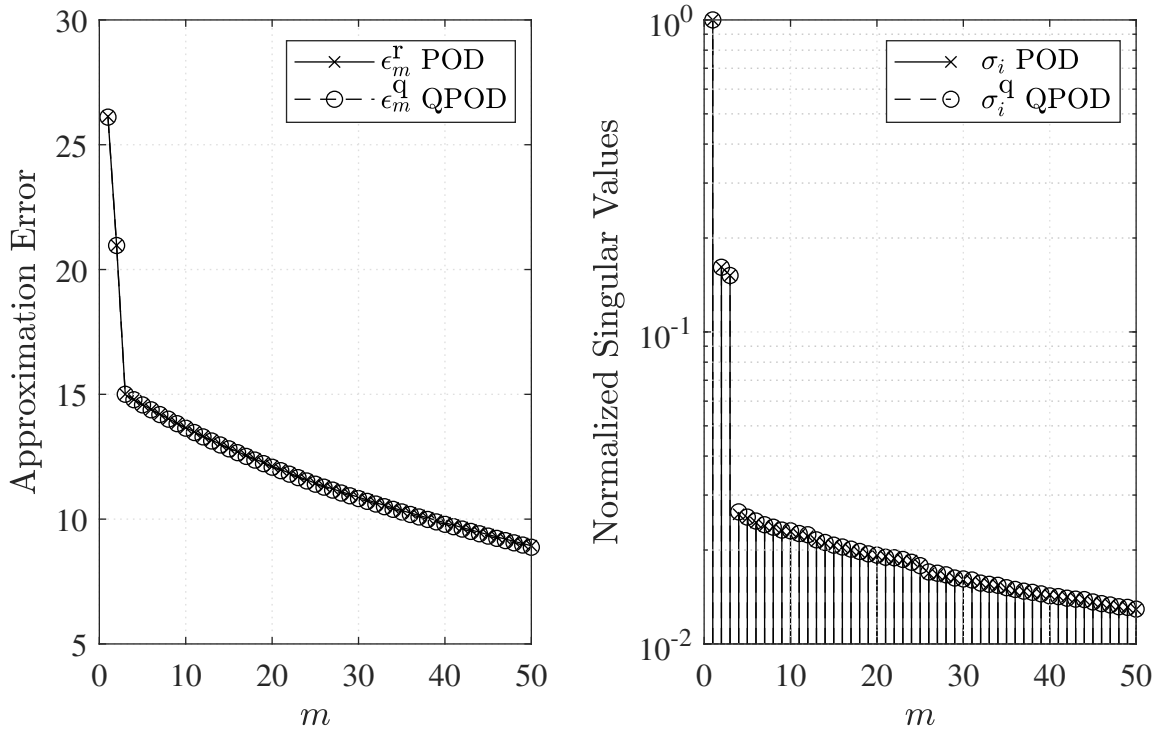
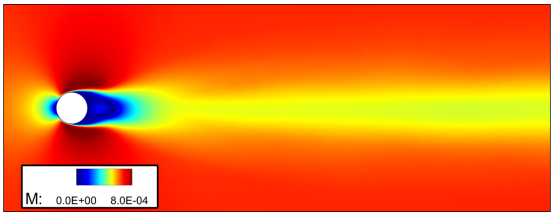
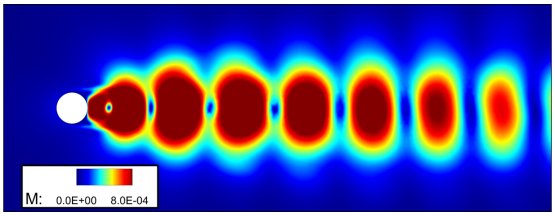


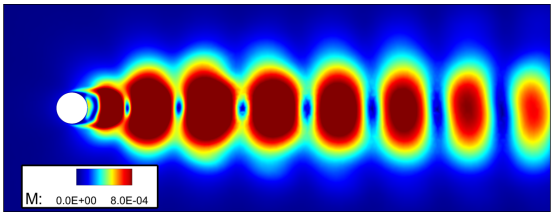
Figure 5.27: Plot to the left shows the ROM errors for the POD and the QPOD methods. The plot on the right shows the normalized singular values associated with each mode from each method. Results are for $Re = 600$.



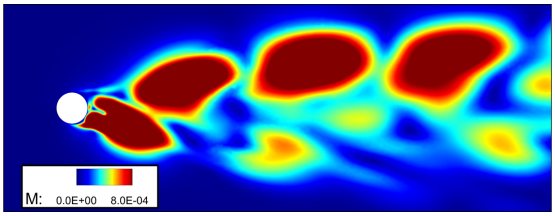
(a) Contours of the velocity magnitude of the average flow. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.



(b) Contours of the velocity magnitude of the first mode, M_1^{r}. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.



(c) Contours of the velocity magnitude of the second mode, M_2^{r}. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.



(d) Contours of the velocity magnitude of the third mode, M_3^{r}. The specific contour plot shown is a planar slice along the Y-axis at Y = 6.

Figure 5.28: Subset of the POD modes generated using the data set with $Re = 600$. The figures show contours of $M_i^{\mathrm{r}} = \sqrt{(\phi_i^u)^2 + (\phi_i^v)^2 + (\phi_i^w)^2}$.

quaternion proper orthogonal decomposition.

Unfortunately, this experiment did not provide any clear advantages for using the QPOD method over the POD method as the results for both methods were identical. Moreover, the method did not provide insight to any parameter dependence that the QPOD method might exhibit. In the next section, some of the mathematical differences between the POD and QPOD methods are highlighted.

5.2 Mathematical Observations

Experiments one, two, and three clearly present a stark difference between the performances of the two methods. Not only do the two methods produce different singular values, but sometimes they produce very different modes. In fact, it was noted in chapter 3, and clearly demonstrated in this chapter, that the QPOD modes contain an extra component that is not present in the POD modes. Moreover, the components of the QPOD modes can be rotated, generating infinitely many modes for every eigenvalue. In the next sections, more mathematical observations are made regarding the ability of each method to extract dynamical features in a flow, and regarding the ability of each method to create rank-m approximations. These observations can help explain why the results shown might have occurred.

5.2.1 Distilling Dynamics

The results of experiments one, two, and three clearly demonstrated that the QPOD method outperformed the POD method. The performances between the two methods was measured using the approximation errors ϵ_m^{q} and ϵ_m^{r}. Observations 15, 16, 25, 30 and 31 clearly indicate that the QPOD method results in better approximations than the POD method.

Observations 23, 26, and 28 clearly indicate that the QPOD singular values descend in a steeper fashion as compared to the POD singular values. Xu et al. [90] made a similar observation regarding the singular values associated with the POD and the QPOD methods

when applied to color images. The authors explained that the reason why the quaternion singular values descend faster is because the QPOD modes contain more information than the POD modes. In fact, in experiment two, it was shown that using only two QPOD modes, a perfect approximation of data set A was achieved while the POD method required five.

Observation 33 notes that the QPOD modes were sharper than the POD modes. Xu et al. [90] also made a similar observation and suggested that the QPOD method better captured the constancy of images as compared to the POD method.

All of these observations suggest that under certain conditions (larger η_{n}), the QPOD modes better capture the relevant information in a given data set than the POD modes. However, in the literature, no clear explanation was found as to why the QPOD modes retain more information than the POD method. In the remainder of this section, a detailed examination of both methods is presented which reveals why the information content of the QPOD modes is richer when compared to their POD counterparts.

Consider the SVD and the QSVD of the matrices $u^{\mathrm{d}} \in \mathbb{R}^{(n\times 3)\times N}$ and $u^{\mathrm{dq}} \in \mathbb{H}^{n\times N}$ respectively as follows

$$\begin{aligned} u^{\mathrm{d}} &= \Phi \Sigma A, \\ u^{\mathrm{dq}} &= \Phi^{\mathrm{q}} \Sigma^{\mathrm{q}} A^{\mathrm{q}}. \end{aligned} \tag{5.18}$$

Using simple matrix algebra, both the POD modes Φ, and the QPOD modes Φ^{q} are placed to one side of the equal sign such that

$$\begin{aligned} \Phi &= u^{\mathrm{d}} A^{\top} \Sigma^{-1}, \\ \Phi^{\mathrm{q}} &= u^{\mathrm{dq}} A^{\mathrm{q}\star} \left(\Sigma^{\mathrm{q}}\right)^{-1}. \end{aligned} \tag{5.19}$$

An expression for the columns of equations 5.19 is given by

$$\begin{aligned} \phi_i &= \mathrm{col}_i(\Phi) = u^{\mathrm{d}}\, \mathrm{col}_i(A) \sigma_i^{-1} \\ \phi_i^{\mathrm{q}} &= \mathrm{col}_i(\Phi^{\mathrm{q}}) = u^{\mathrm{dq}}\, \mathrm{col}_i(A) (\sigma_i^{\mathrm{q}})^{-1} \end{aligned} \tag{5.20}$$

Equations 5.20 states that the POD modes are a linear combination of the snapshot matrix u^{d}, while the the QPOD modes are a linear combination of the quaternion snapshot matrix u^{dq}. More explicitly, using a simple linear algebra interpretation of matrix multiplication (multiplying a matrix and vector is as a linear combination of the columns the matrix), the i^{th} POD mode is given by

$$\begin{aligned}\phi_i &= u^{\mathrm{d}}\,\mathrm{col}_i(A)\sigma_i^{-1} \\ &= \sum_{j=1}^{N} \mathrm{col}_j(u^{\mathrm{d}})\,\frac{A_{ij}}{\sigma_i} \\ &= \mathrm{col}_1(u^{\mathrm{d}})\frac{A_{i1}}{\sigma_i} + \mathrm{col}_2(u^{\mathrm{d}})\frac{A_{i2}}{\sigma_i} + \cdots + \mathrm{col}_N(u^{\mathrm{d}})\frac{A_{iN}}{\sigma_i}.\end{aligned} \tag{5.21}$$

Since, $(A_{ij}/\sigma_i) \in \mathbb{R}$, the POD modes are a linear combination of the snapshots over the scalar field of real numbers. Using the same argument, the i^{th} QPOD mode is given by

$$\begin{aligned}\phi_i^{\mathrm{q}} &= u^{\mathrm{dq}}\,\mathrm{col}_i(A^{\mathrm{q}})(\sigma_i^{\mathrm{q}})^{-1} \\ &= \sum_{j=1}^{N} \mathrm{col}_j(u^{\mathrm{dq}})\,\frac{A_{ij}^{\mathrm{q}}}{\sigma_i^{\mathrm{q}}} \\ &= \mathrm{col}_1(u^{\mathrm{dq}})\frac{A_{i1}^{\mathrm{q}}}{\sigma_i^{\mathrm{q}}} + \mathrm{col}_2(u^{\mathrm{dq}})\frac{A_{i2}^{\mathrm{q}}}{\sigma_i^{\mathrm{q}}} + \cdots + \mathrm{col}_N(u^{\mathrm{dq}})\frac{A_{iN}^{\mathrm{q}}}{\sigma_i^{\mathrm{q}}}.\end{aligned} \tag{5.22}$$

Since, $(A_{ij}^{\mathrm{q}}/\sigma_i^{\mathrm{q}}) \in \mathbb{H}$, the QPOD modes are a linear combination of the snapshots over the division algebra of quaternion numbers.

Observation 35. *The POD modes are formed via linear combinations of the snapshots using real scalars as follows*

$$\phi_i = \sum_{i=1}^{N} col_i(u^d)\alpha_i, \quad \alpha_i \in \mathbb{R}. \tag{5.23}$$

The QPOD modes are formed via linear combinations of the snapshots using quaternion

scalars as follows

$$\phi_i^q = \sum_{i=1}^{N} col_i(u^{dq})\alpha_i^q, \quad \alpha_i^q \in \mathbb{H}. \tag{5.24}$$

Hence, both methods are linear and obey the principle of superposition, but only over their respective algebras.

Observation 36. *Given a data matrix u^d, the POD method can only extract the features present in u^d that can be expressed by addition and multiplication of the columns of u^d. On the other hand, given a quaternion representation of the same data matrix u^{dq}, the QPOD method can only extract the features present in u^{dq} that can be expressed as an addition, multiplication, and rotation of the columns of u^{dq}.*

Observations 35 and 36 imply that the QPOD method has more freedom in the operations it can employ to a data set in order to extract the important features present in that data set. In the context of dynamical systems, this translates to a superior capability of the QPOD method to isolate and distill the essential dynamics present in a data set. This explains why the QPOD modes retain more information as compared to the POD modes. In the next section, the ability of each method to combine the components associated with each mode is examined.

5.2.2 Combining Modes, Creating Approximations

Chapter 2 outlines how the POD method performs a modal decomposition of the flow field variables. A modal decomposition breaks down a function of space and time (the flow field variables $U(X,t)$), into the sum of space functions (POD modes $\phi_i(X)$) and temporal functions (time coefficients, $a_i(t)$). The temporal functions $a_i(t)$ determine how each mode contributes to the overall approximation of the original flow. Thus far, a lot of attention was paid to how the POD modes compare to the QPOD modes, but little to no attention has been paid to the role of the temporal functions $a_i(t)$ and how they generalize in the QPOD method.

Observation 22 notes that even when the average mode of a data set was used to create approximations, the QPOD approximations u_1^{dq} were better than the POD approximations u_1^{d}. This observation is surprising because the average POD mode contains exactly the same information as the average QPOD mode. For a given data set, the only difference between the average POD mode and the average QPOD mode is merely a difference of representation. In the case of the POD method, the average mode is represented using real numbers. While in the case of the QPOD method the average mode is represented using quaternion numbers.

Since the average POD mode is the same as the average QPOD mode, but the resulting approximations are different, then the process of combining the modes must be the cause of this disparity. The remainder of this section explores why sometimes the QPOD approximations are superior to the POD approximations, even when the modes are the same. The following treatment borrows several ideas and observations from the work of Xu et al. [90].

In order to understand why the QPOD approximations are better than the POD approximations, both methods are reformulated in three different forms:

1. Summation form
2. Equation form
3. Matrix form

These reformulations are equivalent to one another, but they greatly contrast the differences between the components associated with the POD modes and the components associated with QPOD modes. All three mentioned forms are defined for both the POD method and QPOD method and presented next.

The following is a representation of the POD method and the QPOD method in their

summations forms. The summation forms of the POD and QPOD methods are given by

$$u^{\mathrm{d}} = \sum_{i=1}^{N} \phi_i a_i, \tag{5.25}$$

$$u^{\mathrm{dq}} = \sum_{i=1}^{N} \phi_i^{\mathrm{q}} a_i^{\mathrm{q}}. \tag{5.26}$$

The POD summation form can be expanded as follows

$$u^{\mathrm{d}} = \begin{bmatrix} U^{\mathrm{d}} \\ V^{\mathrm{d}} \\ W^{\mathrm{d}} \end{bmatrix} = \sum_{i=1}^{N} \begin{bmatrix} \phi_i^u \\ \phi_i^v \\ \phi_i^w \end{bmatrix} a_i. \tag{5.27}$$

Hence, the summation form can also be rewritten as a set of equations and is termed the equation form of the POD method and is given as follows

$$\begin{aligned} U^{\mathrm{d}} &= \Phi^u a, \\ V^{\mathrm{d}} &= \Phi^v a, \\ W^{\mathrm{d}} &= \Phi^w a. \end{aligned} \tag{5.28}$$

where $\Phi^{\mathrm{u}} = [\phi_1^{\mathrm{u}} \dots \phi_N^{\mathrm{u}}]$, $\Phi^{\mathrm{v}} = [\phi_1^{\mathrm{v}} \dots \phi_N^{\mathrm{v}}]$, $\Phi^{\mathrm{w}} = [\phi_1^{\mathrm{w}} \dots \phi_N^{\mathrm{w}}]$, and $a = [a_1 \dots a_N]$.

The QPOD summation form can also be expanded as follows

$$\begin{aligned} u^{\mathrm{dq}} &= U^{\mathrm{d}}\boldsymbol{i} + V^{\mathrm{d}}\boldsymbol{j} + W^{\mathrm{d}}\boldsymbol{k} \\ &= \sum_{i=1}^{N} \left(\phi_i^{rq} + \phi_i^{iq}\boldsymbol{i} + \phi_i^{jq}\boldsymbol{j} + \phi_i^{kq}\boldsymbol{k}\right)\left(a_i^{\mathrm{rq}} + a_i^{\mathrm{iq}}\boldsymbol{i} + a_i^{\mathrm{jq}}\boldsymbol{j} + a_i^{\mathrm{kq}}\boldsymbol{k}\right). \end{aligned} \tag{5.29}$$

Let $\Phi^{\mathrm{q}} = [\phi_1^{\mathrm{q}} \dots \phi_N^{\mathrm{q}}]$ and $a^{\mathrm{q}} = [a_1^{\mathrm{q}} \dots a_N^{\mathrm{q}}]$, then by carefully expanding the terms and carrying out the necessary quaternion multiplications, the following set of equations which

are termed the equation form the QPOD method are obtained

$$\begin{aligned} 0 &= \Phi^{\mathrm{rq}} a_0 - \Phi^{\mathrm{iq}} a_1 - \Phi^{\mathrm{jq}} a_2 - \Phi^{\mathrm{kq}} a_3, \\ U^{\mathrm{d}} &= \Phi^{\mathrm{rq}} a_1 + \Phi^{\mathrm{iq}} a_0 + \Phi^{\mathrm{jq}} a_3 - \Phi^{\mathrm{kq}} a_2, \\ V^{\mathrm{d}} &= \Phi^{\mathrm{rq}} a_2 - \Phi^{\mathrm{iq}} a_3 + \Phi^{\mathrm{jq}} a_0 + \Phi^{\mathrm{kq}} a_1, \\ W^{\mathrm{d}} &= \Phi^{\mathrm{rq}} a_3 + \Phi^{\mathrm{iq}} a_2 - \Phi^{\mathrm{jq}} a_1 + \Phi^{\mathrm{kq}} a_0, \end{aligned} \tag{5.30}$$

where $\Phi^{\mathrm{rq}} = \mathrm{real}(\Phi^{\mathrm{q}})$, $\Phi^{\mathrm{iq}} = \mathrm{imag}_{\boldsymbol{i}}(\Phi^{\mathrm{q}})$, $\Phi^{\mathrm{jq}} = \mathrm{imag}_{\boldsymbol{j}}(\Phi^{\mathrm{q}})$, $\Phi^{\mathrm{kq}} = \mathrm{imag}_{\boldsymbol{k}}(\Phi^{\mathrm{q}})$, and $a_0 = \mathrm{real}(a^{\mathrm{q}})$, $a_1 = \mathrm{imag}_{\boldsymbol{i}}(a^{\mathrm{q}})$, $a_2 = \mathrm{imag}_{\boldsymbol{j}}(a^{\mathrm{q}})$, $a_3 = \mathrm{imag}_{\boldsymbol{k}}(a^{\mathrm{q}})$.

The equation forms of the POD and QPOD methods are linear, hence they can be expressed in terms of matrix multiplication. This leads to the following matrix form of the POD method

$$\begin{bmatrix} 0, & U^{\mathrm{d}}, & V^{\mathrm{d}}, & W^{\mathrm{d}} \end{bmatrix} = \begin{bmatrix} 0, & \phi^u, & \phi^v, & \phi^w \end{bmatrix} \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & a \end{bmatrix}, \tag{5.31}$$

and the matrix form of the QPOD method

$$\begin{bmatrix} 0, & U^{\mathrm{d}}, & V^{\mathrm{d}}, & W^{\mathrm{d}} \end{bmatrix} = \begin{bmatrix} \phi^{\mathrm{rq}}, & \phi^{\mathrm{iq}}, & \phi^{\mathrm{jq}}, & \phi^{\mathrm{kq}} \end{bmatrix} \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ -a_1 & a_0 & -a_3 & a_2 \\ -a_2 & a_3 & a_0 & -a_1 \\ -a_3 & -a_2 & a_1 & a_0 \end{bmatrix}. \tag{5.32}$$

Table 5.3 summarizes the different representations of the POD and QPOD methods.

	POD	**QPOD**
	$\phi_i^u = \text{row}_{j=1}^{n}(\phi_i)$ $\phi_i^v = \text{row}_{j=n+1}^{n\times 2}(\phi_i)$ $\phi_i^w = \text{row}_{j=n\times 2+1}^{n\times 3}(\phi_i)$	$\phi_i^{\text{rq}} = \text{real}(\phi_i^{\text{q}})$ $\phi_i^{\text{iq}} = \text{imag}_{\boldsymbol{i}}(\phi_i^{\text{q}})$ $\phi_i^{\text{jq}} = \text{imag}_{\boldsymbol{j}}(\phi_i^{\text{q}})$ $\phi_i^{\text{kq}} = \text{imag}_{\boldsymbol{k}}(\phi_i^{\text{q}})$
Summation Form	$u^{\text{d}} = \sum_{i=1}^{N} \phi_i a_i$	$u^{\text{dq}} = \sum_{i=1}^{N} \phi_i^{\text{q}} a_i^{\text{q}}$
Equation Form	$U^{\text{d}} = \Phi^{\text{u}} a$ $V^{\text{d}} = \Phi^{\text{v}} a$ $W^{\text{d}} = \Phi^{\text{w}} a$	$0 = \Phi^{\text{rq}} a_0 - \Phi^{\text{iq}} a_1 - \Phi^{\text{jq}} a_2 - \Phi^{\text{kq}} a_3$ $U^{\text{d}} = \Phi^{\text{rq}} a_1 + \Phi^{\text{iq}} a_0 + \Phi^{\text{jq}} a_3 - \Phi^{\text{kq}} a_2$ $V^{\text{d}} = \Phi^{\text{rq}} a_2 - \Phi^{\text{iq}} a_3 + \Phi^{\text{jq}} a_0 + \Phi^{\text{kq}} a_1$ $W^{\text{d}} = \Phi^{\text{rq}} a_3 + \Phi^{\text{iq}} a_2 - \Phi^{\text{jq}} a_1 + \Phi^{\text{kq}} a_0$
Matrix Form	Equation 5.31	Equation 5.32

Table 5.3: A summary highlighting how the components associated with the POD mode and the components associated with the QPOD modes contribute to the reconstruction of a data set.

All three forms are equivalent to one another, however, by inspecting the different forms, a number of observations can be made

Observation 37. *The equation form reveals that the POD method creates approximations using the same time coefficient for all the of the components associated with a POD mode. On the other hand, the QPOD method creates approximations using four time coefficients, which allows for a greater flexibility in combining the components of each mode.*

Observation 38. *All of the component of every QPOD mode $\Phi^{rq}, \Phi^{iq}, \Phi^{jq}$, and Φ^{kq} contribute to approximating the data sets U^d, V^d, and W^d. On the other hand, the POD method approximates U^d using only Φ^u, V^d using only Φ^v, and W^d using only Φ^w.*

Observation 39. *The QPOD method results in an extra component associated with every QPOD mode, real(ϕ_i^q). Hence, the QPOD method produces approximations using modes with four components, rather than three, as is the case for the POD method.*

Observation 40. *The equation form of the QPOD method imposes an explicit linear relation between the components of the QPOD modes given by equation 5.30, and is repeated here*

$$0 = \Phi^r a_0 - \Phi^u a_1 - \Phi^v a_2 - \Phi^w a_3. \tag{5.33}$$

Such a relationship is not imposed on the POD modes.

Observation 41. *The QPOD time coefficients a_i^q are a generalization of the POD time coefficients a_i. Only when $a_1 = a_2 = a_3 = 0$ do the QPOD method create approximations by combining the components in a fashion similar to the POD method. In such a situation the quaternion time coefficient reduces to*

$$a^q = \begin{bmatrix} a_0 & 0 & 0 & 0 \\ 0 & a_0 & 0 & 0 \\ 0 & 0 & a_0 & 0 \\ 0 & 0 & 0 & a_0 \end{bmatrix}. \tag{5.34}$$

These observations clearly show that the QPOD method is more powerful than the POD method in generating approximations. This is because the QPOD method is more versatile in combining the components associated with its modes in order to generate approximations as compared to the POD method which is more restricted.

A rank-m approximation created via the POD method is given in its summation form as

$$u^{\mathrm{d}} = \sum_{i=1}^{m} \begin{bmatrix} \phi_i^u \\ \phi_i^v \\ \phi_i^w \end{bmatrix} a_i, \tag{5.35}$$

whereas a rank-m approximation created via the QPOD method is given in its summation form as

$$u^{\mathrm{dq}} = \sum_{i=1}^{m} \left(\phi_i^{rq} + \phi_i^{iq}\boldsymbol{i} + \phi_i^{jq}\boldsymbol{j} + \phi_i^{kq}\boldsymbol{k}\right) a_i^{\mathrm{q}}. \tag{5.36}$$

Hence, even when the QPOD modes are the same as the POD modes (i.e. $\phi_i^{\text{rq}} = 0$, $\phi_i^{\text{iq}} = \phi_i^u$, $\phi_i^{\text{jq}} = \phi_i^v$ and $\phi_i^{\text{kq}} = \phi_i^w$), the QPOD method can still produce a better rank-m approximation than the POD method can; because the time coefficients are different and combine the components associated with modes differently. This explains why the QPOD method created better approximations in experiments one, two, and three, even when only the average mode was used to create approximations, as noted in observation 22.

CHAPTER 6
CONCLUSIONS

In 1967, Lumley [1] introduced the proper orthogonal decomposition to the fluid flow community as an attempt to extract persistent, reappearing patterns in turbulent flows. The work by Lumley gained significant attention as it showed promising results [84] in it ability to extract eddies and other features that occur in fluid flow problems [2]. However, it was not until 1987, when Sirovich [3] introduced the snapshot implementation, that the POD receive widespread attention, elevating the POD to a benchmark method for the modal analysis and reduced-order modeling of fluid systems.

Since then, much work has been done to combat the limitations associated with the POD. A great limitation of the POD method is due to the linearity of the method [2], which was detailed in section 5.2.1. When the flow is complex, the linear procedure utilized by the POD method results in modes that do not capture the essence of such a flow. In these cases, the POD modes are not easy to interpret and visualizing the POD modes does not add insight to the understanding of the underlying physics of the flow. Moreover, under these circumstances, the reduced-order models created require a large number of modes in order to correctly approximate the original system. Some of the recent contributions to address these issues are summarized by Rowley et al. [30] and Taira et al. [4].

To the knowledge of the author, this thesis is the first to incorporate quaternions into Lumley's framework for the purpose of extracting patterns from three-dimensional fluid systems. The introduction of quaternions into Lumley's mathematical framework extends the proper orthogonal decomposition to the quaternion proper orthogonal decomposition. Using a quaternion variable to describe the three velocity components of a flow provides a natural means to integrate the flow field variables as a single holistic variable. In addition, quaternions address the three-dimensional nature of fluid problems which is not elegantly

captured by the POD. In fact, merely using a quaternion representation can be considered as a model reduction technique as it reduces the number of variables by two thirds. This lead to the research objective of this thesis, restated here as follows:

Research Objective. *Explore the use of quaternions in the context of modal analysis and reduced-order modeling of three-dimensional fluid systems.*

Since this work is the first of its kind, the scope of the thesis was narrowed down to investigate how a quaternion approach to the benchmark method (POD) would compare. Prior work has investigated the QPOD method, more notably the work of Le Bihan [89] in three-dimensional signal processing and the work of Xu et al. [90] in color image compression. However, no research has been done to investigate the application of QPOD to fluid problems, even though a quaternion formulation of the Navier-Stokes equations was derived by Gibbon [106] and later reformulated by Postnikov and Stepanova [107]. The purpose of this thesis was to investigate the benefits of using quaternions in the context of modal analysis and reduced-order modeling of three-dimensional fluid systems. This was formulated as the main research question, research question 1, of this thesis, which is repeated here as follows:

Research Question 1. *How would a quaternion approach to the POD method in the context of modal analysis and reduced-order modeling of three-dimensional fluid systems compare to the traditional POD method?*

Four experiments were set up to provide the necessary evidence needed to answer this question. All of the four experiments applied the POD and the QPOD methods to different data sets, and outcomes of each method were then compared and contrasted with one another. The first experiment was designed to test the average performance of both methods and to discover any correlations that might be present. Hence, one thousand data sets consisting of three matrices $U^{\mathrm{d}}, V^{\mathrm{d}}, W^{\mathrm{d}} \in \mathbb{R}^{5\times 5}$ were generated. The data sets were randomly generated due to the difficulty in obtaining and applying the POD and QPOD

methods to a thousand data sets pertaining to a three-dimensional fluid problem. The results show clearly that the rank-m approximations generated using the QPOD method were better than the rank-m approximations generated using the POD method. Moreover, strong correlations were also uncovered, most notably, a correlation between the distance metric η and the performance metrics used.

This correlation suggested that the differences between the POD and QPOD methods can be further exaggerated. This was tested in the second experiment, where two data sets were generated, data set A and data set B. Data set A was generated using `MATLAB`'s optimization routine such that η was maximized, whereas data set B was generated such that η was minimized. The results where surprising as the QPOD showed a huge performance advantage over the POD method when applied to data set A but no difference in results when applied to data set B. These finding further confirm the correlation between η and the performance metrics.

Until then, the methods have still not been applied to a data set pertaining to a three-dimensional fluid system. Thus, in experiment three, the methods were applied to a data set obtained by Eppink [112] at NASA Langley. The data set pertained to measurement of a three-dimensional fluid flow over a forward facing step collected at six locations downstream of the flow. The POD and QPOD methods were then applied to the data sets. The results consistently verified that the QPOD method was better than the POD method.

Two other research questions were also formulated and addressed in this thesis. Namely, research question 2, which discusses the scalability of the QPOD method restated here as follows:

Research Question 2. *Is the quaternion proper orthogonal decomposition scalable?*

In order to address these questions, experiment four was carried out. In experiment four, a numerical model of the flow around a three-dimensional cylinder was simulated at Reynolds numbers 35, 120, 240, 300, and 600. The flow around a three-dimensional cylinder was chosen because such a flow has been been rigorously investigated in the past and

contains a number of rich, parameter-dependent phenomenons, particularly, a sequence of period doubling that transitions the flow from laminar to turbulence. The grid generated to simulate the flow comprised of 2,271,775 nodes, simulations of the grid produced moderately large data sets; around 48 Gigabytes of memory for each data set. Inspired by the work of Sirovich [3], the quaternion snapshot implementation was defined for the first time in this thesis, and successfully applied the QPOD method to all the data sets. The results of the quaternion snapshot implementation were also verified to be accurate as well. Unfortunately, the results of the experiment did not produce any evidence to show any Reynolds number dependence on the performance of the QPOD method. Moreover, the results of the POD and the QPOD methods were identical as η was negligible.

In short, the majority of the results clearly show an added advantage of the QPOD method over the POD method. The results of all the experiments indicate that for a given data set, larger values of η_{n} indicate that rank-m approximations generated via the QPOD method are more accurate than the POD method. In the worse can scenario, $\eta_n = 0$, the QPOD method and the POD method produce identical results. Hence, there are no downsides to a quaternion approach to the POD method other than a slight increase in computational resources. Even then, the quaternion snapshot implementation alleviates this issue, as was demonstrated in experiment four supporting the following claim:

Hypothesis 2. *If the quaternion snapshot implementation is used, then the QPOD method is scalable.*

It was also mathematically shown that the QPOD method can better distill the essential dynamics present in a data set by creating richer modes, which were termed the quaternion coherent structures. Moreover, it was shown that the QPOD method can combine the components associated with its modes in more ways than the traditional POD method can. The extra flexibility of the QPOD method in combining the components associated with QPOD modes results in more accurate rank-m approximations as compared to the rank-m approximations generated by the more restrictive POD method.

These findings provide sufficient evidence to support the main claim of this research, which is restated here as follows:

Hypothesis 1. *If a quaternion approach is used in the context of model-order reduction of three dimensional fluid systems, then the QPOD method will out perform the traditional POD method.*

In summary the three major contributions of this thesis are:

1. A quaternion approach to the POD method was introduced and its performance was assessed in the context of modal analysis and reduced-order modeling of three-dimensional fluid systems.

2. The quaternion snapshot implementation was devised, which allows for the scalability of the QPOD method.

3. A mathematical treatment that compares the POD and QPOD methods was presented.

A quaternion approach provides for a more natural, physics-based framework for treating three-dimensional fluid flow problems while generalizing the proper orthogonal decomposition, hence preserving its favorable features and extending the method to higher dimensional spaces. A quaternion description of three-dimensional fluid systems is a more sophisticated and a appropriate mathematical representation, which on a fundamental level abstracts Lumley's mathematical representation of coherent structures to what is defined for the first time in this thesis as the quaternion coherent structures.

It appears that quaternion coherent structures contain information that is dynamically richer than their non-quaternion counterpart, which achieves the following goals:

1. Provides a more informative modal decomposition of fluid systems. This helps scientists and engineers further their understanding of fluid flow.

2. Produces more efficient reduced-order models that more accurately capture flow dynamics and interactions between the flow field variables.

Furthermore, the quaternion coherent structures exhibit an interesting phenomenon which was termed in this thesis as the kaleidoscope effect. The kaleidoscope effect occurs because the components of the quaternion coherent structures can rotate, while remaining an eigenvector of the quaternion two point spatial correlation matrix **??**. When the POD modes are identical to the QPOD modes, the POD modes can be cast into a quaternion form, which allows for the rotation of the POD modes as well. This suggests that the quaternion coherent structures are an abstraction and a generalization of the coherent structures defined by Lumley [1], and not just an arbitrary definition. This also suggests that the coherent structure are only a slice of the full picture that can be observed using the quaternion coherent structures. The kaleidoscope effect can be experienced by continuously rotating the components of the quaternion coherent structures. A visualization tool for the quaternion coherent structures of experiment three is written in `MATLAB` and available online at `https://github.com/yissac/kaleidoscope.git`.

The consequences of such an improved modal analysis and reduced-order modeling capability of fluid systems will greatly aid in many aspects of aircraft design, particularly in the aeroelastic analysis and design. The QPOD method provides a superior capability to capture, isolate, and distill the complex aerodynamics over flexible structures encountered in modern day and future aircraft designs. These accurate but lower-order representations will pave the way for surrogate-based optimization, uncertainty quantification, and fluid flow control.

6.1 Limitations

In experiment one, observation 14 it was shown that the distance metric varies from data set to data set. Moreover, a strong relationship was identified between the distance metric η and the performance measure ϵ_m. Observations 19, 17, and 18 indicate that the performance of

the QPOD method is compromised for smaller values of η. Experiment four applied both the POD and the QPOD methods to the flow evolving around a three-dimensional cylinder. As detailed in the chapter 4, the flow was simulated with FUN3D using the incompressible Navier-Stokes equations. The results of experiment four showed identical results between the POD and the QPOD where $\eta \approx 0$.

It is still unclear why both methods produced identical results, but a good guess is that it could be due to the incompressibility condition. Keefe and Moin [41] state that the incompressibility of the flow implies that only two of the velocity components are independent. In such a case, the higher dimensional nature of quaternions is not fully exploited which might explain why both methods produced identical results. However, this does present an opportunity for future work, namely, further investigation should examine how different assumptions, flow parameters, etc., have on the performance of the QPOD method.

Experiment two revealed that when η was maximized the POD method performed poorly, and that the singular values associated with the POD method were all equal 5.5a. In 1991, Aubry [133] introduced the concepts of global energy and global entropy from probability theory to dynamical system. Aubry's definition of global entropy is the following

$$H = \frac{1}{\log N} \sum_{k=1}^{N} p_k \log(p_k), \tag{6.1}$$

where p_k is the ratio between the eigenvalue and the global energy (sum of the non-zero eigenvalues). According to Aubry, this definition of entropy represents the distribution of energy among the eigenvalues. Hence, if the energy is equally distributed among the modes, then the eigenvalues are equal, and the entropy takes a value of 1. This was exactly the situation observed in experiment two for data set A, seen in figure 5.5a, which indicates a possible relationship between η and the entropy defined by Aubry.

Finally, it is important to compare the computational costs of the POD and QPOD methods. Using careful book-keeping, the multiplication of two quaternions requires 28

floating-point operations (FLOPs). However, when the quaternions are pure (zero real part), which is the case in the work of this thesis, their product requires only 14 FLOPs. Depending on the implementation of the SVD and QSVD being used, Li et al. [108] notes that given a data set $u^{\mathrm{d}} \in \mathbb{R}^{3n\times N}$, the SVD requires $12nN^2 + 9N^3$ FLOPs, while it would require the QVSD $96nN^2 + \frac{64}{3}N^3$ FLOPs when applied to the quaternion representation of the same data set $u^{\mathrm{dq}} \in \mathbb{H}^{n\times N}$. Hence, for a data set with N snapshots and n data points per snapshot, the complexity of the POD is $\mathcal{O}\left(9N^3\right)$ while the complexity of the QPOD is $\mathcal{O}\left(\frac{64}{3}N^3\right)$. The analysis shows that the computational complexity of both algorithms are comparable, however, as N becomes larger, the computational speed of the POD over the QPOD will become more noticeable.

However, if the QPOD method is applied in a situation where η is large, reduced-order models created via the QPOD method will be more efficient and hence run faster than the POD reduced-order models. Moreover, applying the quaternion snapshot implementation of the QPOD method can alleviate some of the computational burden associated with QSVD.

6.2 Future Work

The work presented in this thesis is a first attempt at incorporating quaternions for the purpose of model-order reduction of three-dimensional fluid systems. The findings presented in the work of this thesis raise a lot of interesting questions and opens new areas of research.

It was shown that the QPOD generalizes the concept of coherent structures to a more abstract set of patterns, the quaternion coherent structures. It was shown that such structures contain four components and exhibit a kaleidoscope effect. Future research should explore the different interpretations associated with the quaternion coherent structures and their implications. The ability to rotate the QPOD modes seems to resemble the rotations of the bases used to represent the infinite realizations of a dynamical system. One such interpretation might be that QPOD modes generalize over the coordinates of a system and

hence they are coordinate free, even though the QPOD modes were computed using a data set with a coordinate system.

By applying the QPOD method to different fluid systems at different flow regimes, future research can exploit any hidden potential associated with a quaternion approach and conceptually further the current understanding of fluid flows. For example, it would be interesting to see what insights could be gained by applying the QPOD method to flows with shocks. This also includes incorporating more flow field variables such as pressure, density and energy into a quaternion description that would be more suitable for different types of flow such as compressible flow. In fact, it would be interesting to incorporate all the flow field variables into a single holistic variable by looking beyond the quaternion algebra and into the eight dimensional algebra known as octonions.

In 1991, Breuer and Sirovich [85] demonstrated the robustness of the POD method in its ability to extract the basis functions from simulation data under different resolutions and noise contamination schemes. Future work should also compare the performance of the POD and QPOD methods under different resolutions and noise contamination schemes.

In section 3.2.1, a quaternion formulation of the Navier-Stokes equations was presented. Future research should explore projection-based model reduction (Galerkin and Petrov-Galerkin approximations) of the quaternion coherent structures on the quaternion formulation of the Navier-Stokes equations. Since the quaternion coherent structures can be richer than their non quaternion counterpart, projecting the dynamics of fluid systems onto their span might yield more efficient and more accurate reduced-order models.

In addition, future work should also explore the potential application of the QPOD method in areas where the POD method has been applied, such as aerodynamic shape optimization, uncertainty quantification, and controls. Moreover, future research should adapt and extend the QPOD method by applying the same ideas previously developed to adapt and extend the POD method. For example, the ideas behind the BPOD, SPOD, TPOD, Gappy POD, DEIM, etc., may cross fertilize into the QPOD.

Recent developments in the field of quaternion-valued random variables also seem to harness great potential. Advances like the quaternion uncorrelating transform (QUT) completely exploit second-order statistics associated with the flow. Such a transform not only uncorrelates the different snapshots, but also seeks to uncorrelate the flow field variables as well.

Another area for future work should investigate the parametric adaptation of quaternion coherent structures for different parameter ranges. In such a fashion, quaternions could be used to interpolate between reduced-order models created at for different flow parameters.

The results presented in this thesis provide compelling evidence advocating for the use of quaternions in the context of modal analysis and reduced-order modeling of fluid systems. Future work should therefore continue to explore and exploit the use of quaternions as that opens up uncharted research areas. These discoveries will revolutionize the current state of the art and might bring about a new understandings of fluid flow.

Appendices

APPENDIX A

QUATERNION ALGEBRA

A.1 Aleternate Proof to Equation 3.60

Let $u^{\text{dq}} \in \mathbb{H}^{n\times N}$ be a pure quaternion matrix, such that all the entries of the quaternion matrix u^{dq} have zero real part, $q_{ij} = (0, u_{ij}), 1 < i < n, 1 < j < N$,

$$u^{\text{dq}} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1N} \\ q_{21} & q_{22} & \cdots & q_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n1} & q_{n2} & \cdots & q_{nN} \end{bmatrix}.$$

Consider the following

$$u^{\text{dq}}u^{\text{dq}\star} = \begin{bmatrix} \sum_i^N q_{1i}q_{1i}^\star & \sum_i^N q_{1i}q_{2i}^\star & \cdots & \sum_i^N q_{1i}q_{ni}^\star \\ \sum_i^N q_{2i}q_{1i}^\star & \sum_i^N q_{2i}q_{2i}^\star & \cdots & \sum_i^N q_{2i}q_{ni}^\star \\ \vdots & \vdots & \ddots & \vdots \\ \sum_i^N q_{ni}q_{1i}^\star & \sum_i^N q_{ni}q_{2i}^\star & \cdots & \sum_i^N q_{ni}q_{ni}^\star \end{bmatrix}.$$

Expanding the quaternion multiplication results in the following expression

$$u^{\text{dq}}u^{\text{dq}\star} = \begin{bmatrix} \sum_i^N (\vec{u}_{1i}\cdot\vec{u}_{1i}, \vec{u}_{1i}\times -\vec{u}_{1i}) & \cdots & \sum_i^N (\vec{u}_{1i}\cdot\vec{u}_{ni}, \vec{u}_{1i}\times -\vec{u}_{ni}) \\ \sum_i^N (\vec{u}_{2i}\cdot\vec{u}_{1i}, \vec{u}_{2i}\times -\vec{u}_{1i}) & \cdots & \sum_i^N (\vec{u}_{2i}\cdot\vec{u}_{ni}, \vec{u}_{2i}\times -\vec{u}_{ni}) \\ \vdots & \ddots & \vdots \\ \sum_i^N (\vec{u}_{ni}\cdot\vec{u}_{1i}, \vec{u}_{ni}\times -\vec{u}_{1i}) & \cdots & \sum_i^N (\vec{u}_{ni}\cdot\vec{u}_{ni}, \vec{u}_{ni}\times -\vec{u}_{ni}) \end{bmatrix}. \quad \text{(A.1)}$$

The real and imaginary parts of $u^{\mathrm{dq}}u^{\mathrm{dq}\star}$ are then expressed as follows

$$\begin{aligned}\mathrm{real}(u^{\mathrm{dq}}u^{\mathrm{dq}\star}) &= \begin{bmatrix} \sum_i^N \vec{u}_{1i}\cdot\vec{u}_{1i} & \cdots & \sum_i^N \vec{u}_{1i}\cdot\vec{u}_{ni} \\ \sum_i^N \vec{u_{2i}}\cdot\vec{u_{1i}} & \cdots & \sum_i^N \vec{u}_{2i}\cdot\vec{u}_{ni} \\ \vdots & \ddots & \vdots \\ \sum_i^N \vec{u}_{ni}\cdot\vec{u}_{1i} & \cdots & \sum_i^N \vec{u}_{ni}\cdot\vec{u}_{ni} \end{bmatrix}, \\ &= U^{\mathrm{d}}U^{\mathrm{d}\top} + V^{\mathrm{d}}V^{\mathrm{d}\top} + W^{\mathrm{d}}W^{\mathrm{d}\top}, \\ &= A. \end{aligned} \tag{A.2}$$

$$\begin{aligned}\mathrm{imag}_{\boldsymbol{i}}(u^{\mathrm{dq}}u^{\mathrm{dq}\star}) &= \begin{bmatrix} \sum_i^N w_{1,i}v_{1,i} - v_{1,i}w_{1,i} & \cdots & \sum_i^N w_{1,i}v_{n,i} - v_{1,i}w_{n,i} \\ \sum_i^N w_{1,i}v_{2,i} - v_{2,i}w_{1,i} & \cdots & \sum_i^N w_{2,i}v_{n,i} - v_{2,i}w_{n,i} \\ \vdots & \ddots & \vdots \\ \sum_i^N w_{n,i}v_{1,i} - v_{n,i}w_{1,i} & \cdots & \sum_i^N w_{n,i}v_{n,i} - v_{n,i}w_{n,i} \end{bmatrix}, \\ &= W^{\mathrm{d}}V^{\mathrm{d}\top} - V^{\mathrm{d}}W^{\mathrm{d}\top}, \\ &= B. \end{aligned} \tag{A.3}$$

$$\begin{aligned}\mathrm{imag}_{\boldsymbol{j}}(u^{\mathrm{dq}}u^{\mathrm{dq}\star}) &= \begin{bmatrix} \sum_i^N u_{1i}w_{1i} - w_{1i}u_{1i} & \cdots & \sum_i^N u_{1i}w_{ni} - w_{1i}u_{ni} \\ \sum_i^N u_{2i}w_{1i} - w_{2i}u_{1i} & \cdots & \sum_i^N u_{ni}w_{1i} - w_{ni}u_{1i} \\ \vdots & \ddots & \vdots \\ \sum_i^N u_{ni}w_{1i} - w_{ni}u_{1i} & \cdots & \sum_i^N u_{ni}w_{ni} - w_{ni}u_{ni} \end{bmatrix}, \\ &= U^{\mathrm{d}}W^{\mathrm{d}\top} - W^{\mathrm{d}}U^{\mathrm{d}\top}, \\ &= C. \end{aligned} \tag{A.4}$$

$$
\begin{aligned}
\text{imag}_{\boldsymbol{k}}(u^{\text{dq}}u^{\text{dq}\star}) &= \begin{bmatrix} \sum_i^N v_{1,i}u_{1,i} - u_{1,i}v_{1,i} & \cdots & \sum_i^N v_{1,i}u_{n,i} - u_{1,i}v_{n,i} \\ \sum_i^N v_{1,i}u_{2,i} - u_{2,i}v_{1,i} & \cdots & \sum_i^N v_{2,i}u_{n,i} - u_{2,i}v_{n,i} \\ \vdots & \ddots & \vdots \\ \sum_i^N v_{n,i}u_{1,i} - u_{n,i}v_{1,i} & \cdots & \sum_i^N v_{n,i}u_{n,i} - u_{n,i}v_{n,i} \end{bmatrix}, \\
&= V^{\text{d}}U^{\text{d}\top} - U^{\text{d}}V^{\text{d}\top}, \\
&= D.
\end{aligned} \tag{A.5}
$$

Hence,

$$
\text{imag}(u^{\text{dq}}u^{\text{dq}\star}) = \left(W^{\text{d}}V^{\text{d}\top} - V^{\text{d}}W^{\text{d}\top}\right)\boldsymbol{i} + \left(U^{\text{d}}W^{\text{d}\top} - W^{\text{d}}U^{\text{d}\top}\right)\boldsymbol{j} + \left(V^{\text{d}}U^{\text{d}\top} - U^{\text{d}}V^{\text{d}\top}\right)\boldsymbol{k}. \tag{A.6}
$$

Putting the simplified forms of the equations together shows that

$$
\begin{aligned}
u^{\text{dq}}u^{\text{dq}\star} &= A + B\boldsymbol{i} + C\boldsymbol{j} + D\boldsymbol{k} \\
&= \left(U^{\text{d}}U^{\text{d}\top} + V^{\text{d}}V^{\text{d}\top} + W^{\text{d}}W^{\text{d}\top}\right) \\
&\quad + \left(W^{\text{d}}V^{\text{d}\top} - V^{\text{d}}W^{\text{d}\top}\right)\boldsymbol{i} \\
&\quad + \left(U^{\text{d}}W^{\text{d}\top} - W^{\text{d}}U^{\text{d}\top}\right)\boldsymbol{j} \\
&\quad + \left(V^{\text{d}}U^{\text{d}\top} - U^{\text{d}}V^{\text{d}\top}\right)\boldsymbol{k}.
\end{aligned} \tag{A.7}
$$

The norm of $u^{\text{dq}}u^{\text{dq}\star}$ is give by

$$
\|u^{\text{dq}}u^{\text{dq}\star}\|_{\text{F}}^2 = \underbrace{\|A\|_{\text{F}}^2}_{\text{dot product information}} + \underbrace{\|C\boldsymbol{i} + D\boldsymbol{j} + E\boldsymbol{k}\|_{\text{F}}^2}_{\text{cross product information}}, \tag{A.8}
$$

APPENDIX B

DATA SETS

B.1 Experiment Two

B.1.1 Data set A

$$U^{\mathrm{d}}_{\max} = \begin{bmatrix} 0.0787 & 0.00803 & -0.0516 & -0.00677 & 0.137 \\ 0.144 & 0.0369 & 0.00836 & 0.15 & 0.213 \\ 0.00708 & 0.0172 & 0.0674 & 0.12 & -0.0144 \\ 0.0123 & -0.0907 & 0.195 & -0.256 & -0.1 \\ 0.0387 & -6.8e-5 & -0.167 & -0.116 & 0.129 \end{bmatrix} \tag{B.1}$$

$$V^{\mathrm{d}}_{\max} = \begin{bmatrix} 0.0592 & 0.0295 & 0.161 & 0.0357 & 0.0272 \\ -0.126 & 0.0686 & 0.25 & 0.11 & 0.0161 \\ -0.166 & 0.0163 & -0.0175 & 0.0348 & -0.02 \\ 0.149 & -0.263 & -0.0731 & 0.104 & -0.18 \\ 0.237 & 0.0515 & 0.142 & -0.0465 & 0.083 \end{bmatrix} \tag{B.2}$$

$$W^{\mathrm{d}}_{\max} = \begin{bmatrix} 0.00448 & -0.0372 & 0.0168 & -0.112 & 0.0668 \\ 0.0349 & -0.206 & 0.0883 & -0.0852 & 0.126 \\ 0.0175 & -0.106 & 0.0432 & 0.0775 & 0.0105 \\ 0.201 & 0.227 & -0.0195 & -0.0489 & -0.207 \\ -0.0634 & 0.0747 & -0.0451 & -0.169 & 0.0712 \end{bmatrix} \tag{B.3}$$

B.1.2 Data set B

$$U^{\mathrm{d}}_{\mathrm{min}} = \begin{bmatrix} 0.154 & 0.0987 & 0.0621 & -0.00279 & -0.0388 \\ -0.195 & 0.0742 & 0.0832 & -0.00799 & -0.0407 \\ -0.0147 & 0.03 & 0.0747 & -0.00754 & -0.0147 \\ -0.12 & -0.044 & -0.211 & 0.161 & 0.0826 \\ 0.0203 & 0.0976 & 0.123 & -0.0412 & 0.0715 \end{bmatrix} \tag{B.4}$$

$$V^{\mathrm{d}}_{\mathrm{min}} = \begin{bmatrix} 0.0112 & 0.0617 & 0.0169 & 0.0453 & -0.0321 \\ 0.342 & 0.0135 & -0.079 & -0.0506 & 0.085 \\ 0.165 & -0.199 & -0.1 & 0.119 & 0.0212 \\ 0.0735 & 0.0606 & 0.103 & -0.0314 & 0.0216 \\ -0.183 & 0.131 & 0.014 & 0.0157 & 0.119 \end{bmatrix} \tag{B.5}$$

$$W^{\mathrm{d}}_{\mathrm{min}} = \begin{bmatrix} 0.268 & 0.0537 & 0.0211 & -0.00505 & -0.232 \\ 0.277 & -0.113 & -0.0807 & 0.0329 & -0.024 \\ -0.0943 & 0.18 & 0.0412 & -0.0882 & 0.178 \\ -0.102 & 0.021 & -0.00946 & 0.0313 & 0.12 \\ -0.254 & -0.0312 & 0.0196 & 0.0678 & 0.23 \end{bmatrix} \tag{B.6}$$

REFERENCES

[1] J. L. Lumley, "The Structure of Inhomogeneous Turbulent Flows", in *Proceedings of the International Colloquium on the Fine Scale Structure of the Atmosphere and its Influence on Radio Wave Propagation*, A. M. Yaglam and V. I. Tatarsky, Eds., Nauk, Moscow, SSSR, 1967.

[2] G. Berkooz, P. Holmes, and J. L. Lumley, "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows", *Annual Reviews of Fluid Mechanics*, vol. 25, no. 1, pp. 539–575, 1993.

[3] L. Sirovich, "Turbulence and the Dynamics of Coherent Structures, Part I-III", *Quarterly of Applied Mathematics*, vol. XLV, no. 3, pp. 561–590, 1987.

[4] K. Taira, S. L. Brunton, S. T. M. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, "Modal Analysis of Fluid Flows: An Overview", *AIAA Journal*, vol. 55, no. 12, pp. 4013–4041, 2017. arXiv: 1702.01453.

[5] W. K. George, "Insight into the Dynamics of Coherent Structures from a Proper Orthogonal Decomposition", in *Internation Seminar on Wall Turbulence*, 1988.

[6] C. Gao and W. Zhang, "Transonic Aeroelasticity: A New Perspective From the Fluid Mode", *Progress in Aerospace Sciences*, vol. 113, 2020.

[7] L. Gipson, Ed., *Research in 2019 Enables Future Aviation Advances*, https://www.nasa.gov/aeroresearch/research-in-2019-enables-future-aviation-advances, 2019.

[8] Lee Obringer, Ed., *Hyperelastic Research/Lightweight Flexible Aircraft*, https://www.nasa.gov/feature/hyperelastic-researchlightweight-flexible-aircraft, 2015.

[9] Lillian Gipson, Ed., *Low-Boom Flight Demonstration Overview*, https://www.nasa.gov/mission_pages/lowboom/mission, 2019.

[10] M. Conner, Ed., *Highly Flexible Wings Tested*, https://www.nasa.gov/centers/armstrong/features/highly_flexible_wings_tested.html, 2017.

[11] R. E. Bartels and A. I. Sayma, "Computational Aeroelastic Modelling of Airframes and Turbomachinery: Progress and Challenges", *Philosophical transactions. Series*

A, Mathematical, physical, and engineering sciences., vol. 365, no. 1859, pp. 2469–2499, 2007.

[12] D. P. Raymer, *Aircraft Design: A Conceptual Approach*, 6th ed. American Institute of Aeronautics and Astronautics, Incorporated, 2018, 1989.

[13] J. D. Anderson, *Aircraft Performance and Design*, 1st ed. McGraw-Hill Education, 2010.

[14] K. G. Bhatia, "Airplane Aeroelasticity: Practice and Potential", *Journal of Aircraft*, vol. 40, no. 6, pp. 1010–1018, 2003.

[15] E. M. Lee-Rausch and J. T. Batina, "Wing Flutter Boundary Prediction Using an Unsteady Euler Aerodynamic Method", NASA, Tech. Rep., 1993.

[16] K. K. Gupta, "Development of a Finite Element Aeroelastic Analysis Capability", *Journal of Aircraft*, vol. 33, no. 5, 1996.

[17] R. B. Melville, S. A. Morton, and D. P. Rizzetta, "Implementation of a Fully-Implicit, Aeroelastic Navier-Stokes Equations", *American Institute of Aeronautics and Astronautics*, 1997.

[18] E. F. Sheta, "Numerical Analysis of F/A-18 Vertical Tail Buffeting", in *American Institute of Aeronautics and Astronautics*, 2001.

[19] C. Farhat, P. Geuzaine, and G. Brown, "Aeroelastic Dynamic Analysis of a Full F-16 Configuration for Various Flight Conditions", *American Institute of Aeronautics and Astronautics*, vol. 41, pp. 363–371, 2003.

[20] W. A. Silva, A. De La Garza, S. Zink, E. G. Bounajem, J. Christopher Johnson, M. Buonanno, M. D. Sanetrik, P. Chwalowski, S. Y. Yoo, and J. Hur, "An Overview of the NASA High Speed ASE Project: Aeroelastic Analysis of a Low-Boom Supersonic Configuration", in *AIAA Scitech Forum*, 2015.

[21] D. N. Mavris, O. Bandte, and D. A. Delaurentisf, "Robust Design Simulation: A Probabilistic Approach to Multidisciplinary Design", *Journal of Aircraft*, vol. 36, no. 1, pp. 298–307, 1999.

[22] R. Yondo, E. Andrés, and E. Valero, "A Review on Design of Experiments and Surrogate Models in Aircraft Real-Time and Many-Query Aerodynamic Analyses", *Progress in Aerospace Sciences*, vol. 96, pp. 23–61, 2018.

[23] D. J. Lucia, P. S. Beran, and W. A. Silva, "Reduced-Order Modeling: New Approaches for Computational Physics", *Progress in Aerospace Sciences*, vol. 40, no. 1-2, pp. 51–117, 2004.

[24] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, "A Survey of Model Reduction Methods for Large-Scale Systems", Tech. Rep., 2000.

[25] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems (Advances in Design and Control)*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2005, ISBN: 0898715296.

[26] W. H. A. Schilders, H. A. Van der Vorst, and J. Rommes, *Model Order Reduction: Theory, Research Aspects and Applications*, 1st ed., January 2008. Springer-Verlag Berlin Heidelberg, 2008, vol. 13, ISBN: 978-3-540-78840-9. arXiv: arXiv:1011.1669v3.

[27] P. Benner, S. Gugercin, and K. E. Willcox, "A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems", *SIAM Review*, vol. 57, no. 4, pp. 483–531, 2015.

[28] P. Benner, M. Ohlberger, A. Cohen, and K. E. Willcox, *Model Reduction and Approximation*, P. Benner, M. Ohlberger, A. Cohen, and K. Willcox, Eds. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2017.

[29] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine Learning for Fluid Mechanics", *Annual Review of Fluid Mechanics*, vol. 52, no. 1, pp. 477–508, 2020. arXiv: 1905.11075.

[30] C. W. Rowley and S. T. Dawson, "Model Reduction for Flow Analysis and Control", *Annual Review of Fluid Mechanics*, vol. 49, no. 1, pp. 387–417, 2017.

[31] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. Kevin Tucker, "Surrogate-Based Analysis and Optimization", *Progress in Aerospace Sciences*, vol. 41, no. 1, pp. 1–28, 2005.

[32] A. I. Forrester and A. J. Keane, *Recent Advances in Surrogate-Based Optimization*, 2009.

[33] B. A. Winther, P. J. Goggin, and J. R. Dykman, "Reduced-Order Dynamic Aeroelastic Model Development and Integration with Nonlinear Simulation", *Journal of Aircraft*, vol. 37, no. 5, pp. 833–839, 1998.

[34] T. Kim, "An Efficient Response-Based Modal Analysis for Dynamic Systems with Multiple Inputs", in *AIAA Structures, Structural Dynamics and Materials Conference and Exhibit*, 2001.

[35] T. Dissertations and M. P. Theses, "Discrete-Time Linear and Nonlinear Aerodynamic Impulse Responses for Efficient CFD Analyses", Ph.D. dissertation, William and Marry, 1997.

[36] L. Ljung, "Perspectives on System Identification", in *IFAC Proceedings Volumes*, 2008, pp. 7172–7184.

[37] Y. Issac, W. A. Silva, and D. N. Mavris, "Reduced Order Modeling of the Pressure Distribution over the AGARD 445.6 Wing", in *AIAA Atmospheric Flight Mechanics Conference*, Kissimmee, Florida, 2018, ISBN: 978-1-62410-525-8.

[38] M. Winter and C. Breitsamter, "Efficient Unsteady Aerodynamic Loads Prediction Based on Nonlinear System Identification and Proper Orthogonal Decomposition", *Journal of Fluids and Structures*, vol. 67, pp. 1–21, 2016.

[39] T. Lieu, C. Farhat, and M. Lesoinne, "Reduced-Order Fluid/Structure Modeling of a Complete Aircraft Configuration", *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 41-43, pp. 5730–5742, 2006.

[40] D. Amsallem and C. Farhat, "Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity", *AIAA Journal*, vol. 46, no. 7, pp. 1803–1813, 2008.

[41] L. Keefe and P. Moin, "The Dimension of Attractors Underlying Periodic Turbulent Poiseuille Flow", *Journal of Fluid Mechanics*, vol. 242, no. 29, pp. 1–29, 1992.

[42] H. Wagner, "Über die Entstchung des Dynamischen Auftriebes von Tragflugeln", *ZAAM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 5, no. 1, pp. 17–35, 1925.

[43] H. G. Küssner, "Zusammenfassender Bericht über den Instationären Auftrieb von Flügeln", *Luftfahrtforschung*, vol. 13, no. 12, pp. 410–424, 1936.

[44] T. Theodorsen, "General Theory of Aerodynamic Instability and the Mechanism of Flutter", *NACA TR 496*, 1934.

[45] W. R. Sears, "Operational Methods in the Theory of Airfoils in Non-Uniform Motion", *Journal of Aeronautical Sciences*, vol. 8, no. 3, 1941.

[46] E. Albano and W. P. Hodden, "A Doublet-Lattice Method for Calculating Lift Distributions on Oscillating Surfaces in Subsonic Flows", *AIAA Journal*, vol. 7, no. 2, pp. 279–285, 1969.

[47] H Ashley and G Zartarian, "Piston Theory - A New Aerodynamic Tool ffor the Aeroelastician", *Composites Structures*, vol. 33, no. 038, pp. 1109–1118, 1956.

[48] P. Geuzaine, G. Brown, and C. Farhat, "Application of a Three-Field Nonlinear Fluid-Structure Formulation to the Prediction of the Aeroelastic Parameters of an F-16 Fighter", *Computers and Fluids*, vol. 32, no. 1, pp. 3–29, 2003.

[49] M. C. Romanowski and E. H. Dowell, "Reduced Order Euler Equations for Unsteady Aerodynamic Flows - Numerical Techniques", in *34th Aerospace Sciences Meeting and Exhibit*, 1996, p. 528.

[50] M. C. Romanowski, "Reduced-Order Unsteady Aerodynamic and Aeroelastic Models Using Karhunen-Loeve Eigenmodes", in *6th Symposium on Multidisciplinary Analysis and Optimization*, 1996, p. 3981.

[51] E. H. Dowell, K. C. Hall, J. P. Thomas, R. Florea, B. I. Epureanu, and J. Heeg, "Reduced Order Models in Unsteady Aerodynamcis", in *40th Structures, Structural Dynamics, and Materials Conference and Exhibit*, 1999, p. 1261.

[52] K. C. Hall, J. P. Thomas, and E. H. Dowell, "Proper Orthogonal Decomposition Technique for Transonic Unsteady Aerodynamic Flows", *AIAA Journal*, vol. 38, no. 10, pp. 11–14, 2000.

[53] D. Amsallem, M. J. Zahr, Y. Choi, and C. Farhat, "Design Optimization Using Hyper-Reduced-Order Models", *Struct Multidisc Optim*, vol. 51, pp. 919–940, 2015.

[54] K. Kunisch and S. Volkwein, "Control of the Burgers Equation by a Reduced-Order Approach Using Proper Orthogonal Decomposition", *Journal of Optimization Theory and Applications*, vol. 102, no. 2, pp. 345–371, 1999.

[55] B. R. Noack, G. Tadmor, and M. Morzyński, "Low-Dimensional Models for Feedback Flow Control. Part I: Empirical Galerkin Models", *2nd AIAA Flow Control Conference*, 2004.

[56] G. Tadmor, B. R. Noack, M. Morzyński, and S. Siegel, "Low-Dimensional Models for Feedback Flow Control. Part II: Control Design and Dynamic Estimation", *2nd AIAA Flow Control Conference*, 2004.

[57] C. Poussot-Vassal and F. Demourant, "Dynamical Medium (Large)-Scale Model Reduction and Interpolation with Application to Aircraft Systems", *AerospaceLab*, no. 4, 2012.

[58] S. Gugercin, A. C. Antoulas, and C. A. Beattie, "H_2 Model Reduction for Large-Scale Linear Dynamical Systems", *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 2, pp. 609–638, 2008.

[59] C. de Villemagne and R. Skelton, "Model Reduction Using a Projection Formulation", in *26th IEEE Conference on Decision and Control*, vol. 26, 1987, pp. 461–466.

[60] K. Carlberg, M. Barone, and H. Antil, "Galerkin v. Least-Squares Petrov–Galerkin Projection in Nonlinear Model Reduction", *Journal of Computational Physics*, vol. 330, pp. 693–734, 2017.

[61] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge university press, 2012.

[62] R. Temam, *Infinite-Dimensonal Dynamical Systems in Mechanics and Physics*, 1998th ed., F. John, J. E. Marsden, and L. Sirovich, Eds. Springer-Verlag, 1988, vol. 68, ISBN: 978-1-4684-0315-2.

[63] N. Aubry, W.-Y. Lian, and E. S. Titi, "Preserving Symmetries in the Proper Orthogonal Decomposition", *SIAM Journal on Scientific Computing*, vol. 14, no. 2, pp. 483–505, 1993.

[64] M. Bergmann, C.-H. Bruneau, and A. Iollo, "Enablers for Robust POD Models", *Journal of Computational Physics*, vol. 228, no. 2, pp. 516–538, 2009.

[65] J.-N Juang and R. S. Pappa, "An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction", *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 620–627, 1985.

[66] R. Everson and L. Sirovich, "Karhunen-Loève Procedure for Gappy Data", *Journal of the Optical Society of America*, vol. 12, no. 8, pp. 1657–1664, 1995.

[67] T. Bui-Thanh, M. Damodaran, and K. E. Willcox, "Aerodynamic Data Reconstruction and Inverse Design Using Proper Orthogonal Decomposition", *AIAA Journal*, vol. 42, no. 8, pp. 1505–1516, 2004.

[68] P. A. Legresley and J. J. Alonso, "Airfoil Design Optimization Using Reduced Order Models Based on Proper Orthogonal Decomposition", in *Fluids 2000 Conference and Exhibit*, 2000, p. 2545.

[69] K. Y. Tang, W. R. Graham, and J. Peraire, "Active Flow Control Using a Reduced Order Model and Optimum Control", Tech. Rep., 1996.

[70] C. W. Rowley, T. Colonius, and R. M. Murray, "Model Reduction for Compressible Flows Using POD and Galerkin Projection", *Physica D: Nonlinear Phenomena*, vol. 189, no. 1-2, pp. 115–129, 2004.

[71] S. Chaturantabut and D. C. Sorensen, "Nonlinear Model Reduction via Discrete Empirical Interpolation", *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2737–2764, 2010.

[72] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, "An 'Emprical Interpolation' Method: Application to Effecient Reduced-Basis Discretization of Partial Differential Equations", *Comptes Rendus Mathematique*, vol. 339, no. 9, pp. 667–672, 2004.

[73] T. Bui-Thanh, K. Willcox, and O. Ghattas, "Model Reduction for Large-Scale Systems with High-Dimensional Parameteric Input Space", *SIAM Journal on Scientific Computing*, vol. 30, no. 6, pp. 3270–3288, 2008.

[74] T. Lieu, C. Farhat, and M. Lesoinne, "POD-Based Aeroelastic Analysis of a Complete F-16 Configuration: ROM Adaptation and Demonstration", in *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2005.

[75] C. W. Rowley, "Model Reduction for Fluids, Using Balanced Proper Orthogonal Decomposition", *International Journal of Bifurcation and Chaos*, vol. 15, no. 3, pp. 997–1013, 2005.

[76] B. C. Moore, "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction", *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 17–32, 1981.

[77] A. C. Antoulas, D. C. Sorensen, and S. Gugercin, "A Survey of Model Reduction by Balanced Truncation and Some New Results", *International Journal of Control*, vol. 77, no. 8, pp. 748–766, 2004.

[78] S. Lall, J. E. Marsden, and S. Glavaški, "Empirical Model Reduction of Controlled Nonlinear Systems", in *IFAC World Congress*, 1999, pp. 473–478.

[79] ——, "A Subspace Approach to Balanced Truncation for Model Reduction of Nonlinear Control Systems", *International Journal of Robust and Nonlinear Control*, vol. 12, no. 6, pp. 519–535, 2002.

[80] K. E. Willcox and J. Peraire, "Balanced Model Reduction via the Proper Orthogonal Decomposition", *AIAA Journal*, vol. 40, no. 11, pp. 2323–2330, 2002.

[81] Z. Ma, S. Ahuja, and C. W. Rowley, "Reduced-Order Models for Control of Fluids using the Eigensystem Realization Algorithm", *Theoretical and Computational Fluid Dynamics*, vol. 25, no. 1-4, pp. 233–247, 2011. arXiv: 0907.1907.

[82] A. Towne, O. T. Schmidt, and T. Colonius, "Spectral Proper Orthogonal Decomposition and its Relationship to Dynamic Mode Decomposition and Resolvent Analysis", *Journal of Fluid Mechanics*, vol. 847, pp. 821–867, 2020.

[83] M. Sieber, K. Oberleithner, and O. C. Paschereit, "Spectral proper orthogonal decomposition", *Journal of Fluid Mechanics*, vol. 792, pp. 798–828, 2016.

[84] H. P. Bakewell and J. L. Lumley, "Viscous Sublayer and Adjacent Wall Region in Turbulent Pipe Flow", *Physics of Fluids*, vol. 10, no. 9, pp. 1880–1889, 1967.

[85] K. S. Breuer and L. Sirovich, "The Use of the Karhunen-Loève Procedure for the Calculation of Linear Eigenfunctions", *Journal of Computational Physics*, vol. 96, no. 2, pp. 277–296, 1991.

[86] W. R. Hamilton, "On Quaternions, or on a New System of Imaginaries in Algebra", *Proceedings of the Royal Irish Academy*, 1844.

[87] S. De Leo, G. Scolarici, and L. Solombrino, "Quaternionic Eigenvalue Problem", *Journal of Mathematical Physics*, vol. 43, no. 11, pp. 5815–5829, 2002. arXiv: 0211063 [math-ph].

[88] D. R. Farenick and B. A. Pidkowich, "The Spectral Theorem in Quaternions", *Linear Algebra and its Applications*, vol. 371, pp. 75–102, 2003.

[89] N. Le Bihan and J. Mars, "Singular Value Decomposition of Quaternion Matrices: A New Tool for Vector-Sensor Signal Processing", *Signal Processing*, vol. 84, no. 7, pp. 1177–1199, 2004.

[90] Y. Xu, L. Yu, H. Xu, H. Zhang, and T. Nguyen, "Vector Sparse Representation of Color Image Using Quaternion Matrix Analysis", *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1315–1329, 2015.

[91] F. Zhang, "Quaternions and Matrices of Quaternions", *Linear Algebra and Its Applications*, vol. 251, pp. 21–57, 1997.

[92] I. Kantor and A. Solodovnikov, *Hypercomplex Numbers An Elementary Introduction to Algebras*. Springer-Verlag New York, 1989.

[93] N. Le Bihan, "The Geometry of Proper Quaternion Random Variables", *arXiv preprint arXiv:1505.06182*, 2015. arXiv: 1505.06182v2.

[94] M. L. Mehta, *Random Matrices*. Elsevier, 1991.

[95] N. Le Bihan, "Traitement alg{\'e}brique des signaux vectoriels: application en s{\'e}paration d'ondes sismiques", Ph.D. dissertation, Grenoble INPG, 2001.

[96] S. J. Sangwine and N. Le Bihan, "Quaternion Singular Value Decomposition Based on Bidiagonalization to a Real or Complex Matrix Using Quaternion Householder

Transformations", *Applied Mathematics and Computation*, vol. 182, no. 1, pp. 727–738, 2006. arXiv: 0603251 [math].

[97] A. Bunse-Gerstner, R. Byers, and V. Mehrmann, "A Quaternion QR Algorithm", *Numerische Mathematik*, vol. 55, no. 1, pp. 83–95, 1989.

[98] N. Le Bihan and S. J. Sangwine, "Jacobi Method for Quaternion Matrix Singular Value Decomposition", *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 1265–1271, 2007.

[99] S. J. Sangwine and N. Le Bihan, "Computing the SVD of a Quaternion Matrix", in *7th IMA Conference on Mathematics in Signal Processing*, 2006.

[100] S. J. Sangwine, *Quaternion Toolbox for Matlab*, 2005.

[101] L. Sirovich and M. J. Kirby, "Low-Dimensional Procedure for the Characterization of Human Faces", *Optical Society of America*, vol. 4, no. 3, pp. 519–524, 1987.

[102] M. J. Kirby and L. Sirovich, "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces", *EEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103–108, 1990.

[103] N. Le Bihan and S. J. Sangwine, "Color Image Decomposition Using Quaternion Singular Value Decomposition", in *International Conference on Visual Information Engineering*, 2003, pp. 113–116.

[104] ——, "Quaternion Principal Component Analysis of Color Images", in *International Conference on Image Processing*, 2003, pp. 809–812, ISBN: 0780377508.

[105] K. I. Kou and Y.-h. Xia, "Linear Quaternion Differential Equations: Basic Theory and Fundamental Results", *Studies in Applied Mathematics*, vol. 141, no. 1, pp. 3–45, 2018. arXiv: arXiv:1510.02224v5.

[106] J. D. Gibbon, "A Quaternionic Structure in the Three-Dimensional Euler and Ideal Magneto-Hydrodynamics Equations", *Physica D: Nonlinear Phenomena*, vol. 166, no. 1-2, pp. 17–28, 2002.

[107] E. B. Postnikov and O. B. Stepanova, "A Variant of the Pure Quaternionic Representation of the Navier-Stokes Equation", *Advances in Applied Clifford Algebras*, vol. 15, no. 1, pp. 67–69, 2005.

[108] Y. Li, M. Wei, F. Zhang, and J. Zhao, "Comparison of Two SVD-Based Color Image Compression Schemes", *PLoS ONE*, vol. 12, no. 3, pp. 1–14, 2017.

[109] S.-C. Pei, J.-H. Chang, and J.-J. Ding, "Quaternion Matrix Singular Value Decomposition and its Applications for Color Image Processing", in *International Conference on Image Processing*, vol. 1, 2003, pp. 805–808, ISBN: 0-7803-7750-8.

[110] P. B. Denton, S. J. Parke, T. Tao, and X. Zhang, "Eigenvectors from Eigenvalues: A Survey of Basic Identity in Linear Algebra", *arXiv preprint arXiv:1908.03795*, 2019. arXiv: 1908.03795v3.

[111] J. L. Eppink, "The Effect of Forward-Facing Steps on Stationary Crossflow Instability Growth and Breakfown", in *AIAA Aerospace Sciences Meeting*, 2018.

[112] ——, "High-Frequency Secondary Instabilities Downstream of a Forward-Facing Step", in *AIAA Scitech 2020 Forum*, 2020, p. 2243.

[113] J. L. Eppink, R. Wlezien, R. King, and M. Choudhari, "Interaction of a Backward-Facing Step and Crossflow Instabilities in Boundary-Layer Transition", *AIAA Journal*, vol. 56, pp. 1–12, 2017.

[114] C. H. Williamson, "Vortex Dynamics in the Cylinder Wake", *Annual Review of Fluid Mechanics*, vol. 28, no. 1, pp. 477–539, 1996.

[115] H. Benard, ".", *Comptes Rendus de l'Académie des Sciences*, vol. 147, pp. 839–842, 1908.

[116] A. Roshko, "On the Drag and Shedding Frequency of Two-Dimensional Bluff Bodies", *NACA-TN-3169*, 1954.

[117] C. H. Williamson, "The Existence of Two Stages in the Transition to Three-Dimensionality of a Cylinder Wake", *Physics of Fluids*, vol. 31, no. 11, pp. 3165–3168, 1988.

[118] H.-Q. Zhang, U. Fey, and B. R. Noack, "On the Transition of the Cylinder Wake", *Physics of Fluids*, vol. 7, no. 4, pp. 779–794, 1995.

[119] M. J. Feigenbaum, "Universal Behavior in Nonlinear Systems", *Physica. D, Nonlinear phenomena*, vol. 7, no. 1-3, pp. 16–39, 1983.

[120] A. G. Tomboulides, G. S. Triantafyllou, and G. E. Karniadakis, "A new mechanism of period doubling in free shear flows", *Physics of Fluids A: Fluid Dynamics*, vol. 4, no. 7, pp. 1329–1332, 1992.

[121] G. E. Karniadakis and G. S. Triantafyllou, *Three-Dimensional Dynamics and Transition to Turbulence in the Wake of Bluff Objects*, 1. 1992, vol. 238, pp. 1–30, ISBN: 0022112092001.

[122] R. Mittal and S. Balachandar, "Effects of Three-Dimensionality on the Lift and Drag of Nominally Two-Dimensional Cylinders", *Physics of Fluids*, vol. 7, no. 8, pp. 1841–1865, 1995.

[123] A. G. Kravchenko and P. Moin, "Numerical Studies of Flow over a Circular Cylinder at ReD=3900", *Physics of Fluids*, vol. 12, no. 2, pp. 403–417, 2000.

[124] N. Kanaris, D. Grigoriadis, and S. Kassinos, "Three Dimensional Flow Around a Circular Cylinder Confined in a Plane Channel", *Physics of Fluids*, vol. 23, no. 6, 2011.

[125] A. E. Deane, I. G. Kevrekidis, G. E. Karniadakis, and S. A. Orszag, "Low-Dimensional Models for Complex Geometry Flows: Application to Grooved Channels and Circular Cylinders", *Physics of Fluids*, vol. 3, no. 10, pp. 2337–2354, 1991.

[126] B. R. Noack and H. Eckelmann, "A Low-Dimensional Galerkin Method for the Three-Dimensional Flow Around a Circular Cylinder", *Physics of Fluids*, vol. 6, no. 1, pp. 124–143, 1994.

[127] B. R. Noack, K. Afanasiev, M. Morzyński, G. Tadmor, and F. Thiele, "A Hierarchy of Low-Dimensional Models for the Transient and Post-Transient Cylinder Wake", *Journal of Fluid Mechanics*, vol. 497, no. 497, pp. 335–363, 2003.

[128] W. Bonhaus, D.L., Anderson, "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids", *Computers & Fluids*, vol. 23, no. 1, pp. 1–21, 1994.

[129] NASA, Ed., *FUN3D Webpage*, `https://fun3d.larc.nasa.gov/`, 2020.

[130] W Anderson, R. Rausch, and D. Bonhaus, "Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids", in *12th Computational Fluid Dynamics Conference*.

[131] V. Vatsa and M. Carpenter, "Higher-Order Temporal Schemes with Error Controllers for Unsteady Navier-Stokes Equations", in *17th AIAA Computational Fluid Dynamics Conference*.

[132] R. Biedron and J. Thomas, "Recent Enhancements to the FUN3D Flow Solver for Moving-Mesh Applications", in *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*.

[133] N. Aubry, "On the Hidden Beauty of the Proper Orthogonal Decomposition", *Theoretical and Computational Fluid Dynamics*, pp. 339–352, 1991.

VITA

Yanal Issac is an American, who was born in Syria, grew up in the United Arab Emirates, and whose ancestry originates from the Northern Caucasus. From a young age, Yanal loved being challenged, whether at school or at sports. He was sixteen when he won the Karate Lightweight Champion of the United Arab Emirates in full-contact Kumite for two consecutive years, and was the youngest competitor to represent the U.A.E. at the World Karate Tournament in Osaka, Japan and the Middle East Tournament in Tehran, Iran.

Eager to build new innovative technologies and pursue a career full of challenges, Yanal earned a Bachelor's degree in Mechanical engineering from Brigham Young University. During his time at BYU, he worked as a teacher assistant, and was an undergraduate research assistant at the Compliant Mechanisms Research Laboratory for two years.

In 2010, he moved to the United Arab Emirates and worked as an offshore field engineer with the National Petroleum Construction Company. He later quit his job to pursue his graduate degree. In 2014, he graduated with a Master's degree in Aerospace Engineering from Georgia Institute of Technology where he was awarded a Presidential Fellowship. Shortly after, he moved to Virginia in order to complete his doctoral degree while being in close collaboration with NASA through the National Institute of Aerospace. Today, Yanal is a practitioner of Muay Thai, has a black belt in Karate, and is determined to attain a black belt in Brazilian Jiu-Jitsu.